

AD-A256 326



12

The Pennsylvania State University
APPLIED RESEARCH LABORATORY
P.O. Box 30
State College, PA 16804

**DESIGNING GABOR FILTERS FOR
TEXTURE SEGMENTATION**

by

D. F. Dunn, W. Higgins, A. Maida, J. Wakeley

DTIC
ELECTE
OCT 14 1992
A D

Technical Report No. TR 92-03
October 1992

Supported by:
Space and Naval Warfare Systems Command

L.R. Hettche, Director
Applied Research Laboratory

Approved for public release; distribution unlimited

92

002

391007

DEFENSE TECHNICAL INFORMATION CENTER
9226765

189108

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of the collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4382, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE October 1992	3. REPORT TYPE AND DATES COVERED
4. TITLE AND SUBTITLE Design Gabor Filters for Texture Segmentation		5. FUNDING NUMBERS
6. AUTHOR(S) D. F. Dunn, W. Higgins, A. Maida, J. Wakeley		
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) The Applied Research Laboratory P.O. Box 30 State College, PA 16804		8. PERFORMING ORGANIZATION REPORT NUMBER TR-92-03
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Space and Naval Warfare Systems Command Department of the Navy Washington, DC 20363-5000		10. SPONSORING/MONITORING AGENCY REPORT NUMBER N00039-88-0051
11. SUPPLEMENTARY NOTES		
12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for Public Release. Distribution Unlimited.		12b. DISTRIBUTION CODE
13. ABSTRACT (Maximum 200 words) Many texture-segmentation schemes use an elaborate bank of filters to decompose a textured image into a joint space/spatial-frequency representation. While these schemes show promise and some analytical work has been done, the relationship between texture differences and the filter configurations required to discriminate them remains largely known. This thesis examines the issue of designing individual filters. Analysis based on mathematically defined texture models shows that applying a properly configured bandpass filter to a textured image produces distinct output discontinuities at texture boundaries. Depending on the type of texture difference and the filter parameters, these discontinuities form one of four characteristic signatures: a step, valley, ridge, or a step change in average local output variation. Accompanying experimental evidence indicates that these signatures are useful for segmenting an image. Initially, a simple 1-D texture model is used to derive the step and valley signatures. This model leads to a simple analytical development providing helpful insight. The 1-D model, however, has certain limitations. For example, the existence of the ridge signature cannot be shown using this model. Consequently, a more general 2-D model is also presented, leading to a more complex but informative analysis.		
14. SUBJECT TERMS Gabor filters, texture segmentation, filters		15. NUMBER OF PAGES 178
		16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT UNCLASSIFIED	18. SECURITY CLASSIFICATION OF THIS PAGE UNCLASSIFIED	19. SECURITY CLASSIFICATION OF ABSTRACT UNCLASSIFIED
20. LIMITATION OF ABSTRACT		

Abstract

Many texture-segmentation schemes use an elaborate bank of filters to decompose a textured image into a joint space/spatial-frequency representation. While these schemes show promise and some analytical work has been done, the relationship between texture differences and the filter configurations required to discriminate them remains largely unknown. This thesis examines the issue of designing individual filters. Analysis based on mathematically defined texture models shows that applying a properly configured bandpass filter to a textured image produces distinct output discontinuities at texture boundaries. Depending on the type of texture difference and the filter parameters, these discontinuities form one of four characteristic signatures: a step, valley, ridge, or a step change in average local output variation. Accompanying experimental evidence indicates that these signatures are useful for segmenting an image. Initially, a simple 1-D texture model is used to derive the step and valley signatures. This model leads to a simple analytical development providing helpful insight. The 1-D model, however, has certain limitations. For example, the existence of the ridge signature cannot be shown using this model. Consequently, a more general 2-D model is also presented, leading to a more complex but informative analysis. In particular, the 2-D analysis indicates those texture characteristics that are responsible for each signature type and leads to detailed filter design criteria. Even the 2-D analysis, though, makes certain simplifying assumptions that lead to inaccuracies in designing filters for nonhomogeneous textures. To overcome this difficulty, an algorithm was developed that determines the "best" filter

parameters for an arbitrary texture pair. The algorithm effectively performs an exhaustive (but efficient) search of the filter parameter space to determine the filter producing the highest quality signature. Signal detection theory is used to provide a measure of signature quality. Although the analyses presented in this study are based on filters derived from Gabor elementary functions, it is the bandpass nature of the filter that is essential; thus, the results apply to bandpass filters in general.

Accession For	
NTIS CRA&I	<input checked="" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By	
Distribution /	
Availability Codes	
Dist	Avail and/or Special
A-1	

DTIC QUALITY INSPECTED **b**

Table of Contents

List of Tables	viii
List of Figures	ix
Acknowledgments	xii
1 Thesis Overview	1
2 Introduction	8
2.1 Early Texture Perception Models	18
2.2 The Local Spatial-Frequency Approach	30
3 Defining the Filter	36
4 1-D Analysis	43
4.1 The 1-D Texture Model	43
4.2 Step Signature	47
4.3 Valley Signature	51
5 2-D Analysis	57
5.1 2-D Texture Model	57
5.2 Characterizing Gabor-Filter Outputs	62
5.2.1 Textures Consisting of Different Texels: Step Signature	62

5.2.2 Textures Consisting of Identical Texels, but Exhibiting a Texture-Phase Difference: Valley and Ridge Signatures	67
5.2.3 Nonuniform Textures	77
6 Gabor-Filter Parameter Guidelines	81
6.1 Guidelines for Selecting Filter Parameters	81
6.1.1 Texels in Two Textured Regions Differ	83
6.1.2 Texel Spacings Differ between Textured Regions	86
6.2 Parameter Constraints for Filter Banks	87
6.2.1 A Constraint on σ	87
6.2.2 Other Parameter Constraints	89
7 Determining Filter Parameters for Nonuniform Textures	94
7.1 Overview of the GFFS Algorithm	95
7.2 Algorithm Implementation Issues	99
7.2.1 Measuring Step-Signature Quality	99
7.2.2 Gabor-Filter Application via Windowed Fourier Transforms	102
7.2.3 Specifying σ_x and σ_y	104
7.2.4 Modified Version of Gabor Filter	106
7.2.5 Other Issues	110
7.3 Previously Proposed Methods for Designing Gabor Filters	112
7.4 Determining Filter Parameters for Other Signature Types	113
8 Results	114
8.1 1-D Results	114

8.2 2-D Results	117
8.2.1 Difference in Texel Type	119
8.2.2 Difference in Texel Orientation	123
8.2.3 Differences in Horizontal and Vertical Texel Spacing	123
8.2.4 Texture-Phase Differences	128
8.2.5 Texel-Spacing Difference between Regions	131
8.2.6 Nonuniform Textures	134
8.2.7 Miscellaneous Texture Examples	148
9 Conclusion	152
A Choosing the Nonlinearity	156
B Overshoot and Undershoot	160
C Estimating Ridge Height	163
D Implementation Details	166
Bibliography	169

List of Tables

5.1	z as a function of α with ψ_{\max} (in degrees) chosen to maximize z	76
6.1	Gabor-filter design criteria for processing a textured image containing two textured regions, $R_1(t_1, \Delta x_1, \Delta y_1)$ and $R_2(t_2, \Delta x_2, \Delta y_2)$	82
8.1	Comparison of actual and predicted Gabor-filter output values for the step signature in Fig. 8.3b	122
8.2	Comparison of actual and predicted Gabor-filter output values for the step signature in Fig. 8.3d	123
C.1	Numerical evaluation of the integral ratio $\alpha = s_{\delta U}/c_{\delta U}$ as a function of the product of parameters δU and σ_x	165

List of Figures

1.1 Examples of two characteristic signature types: (a) step and (b) valley	3
1.2 Example of the ridge signature	4
1.3 Textured regions exhibiting two types of texture difference	5
1.4 Example of signature exhibiting a step change in average local output variation	7
2.1 Schematic representation of three differently textured regions	9
2.2 Surface orientation perceived due to a texture gradient	11
2.3 Textures with randomly oriented texels	12
2.4 Segmentation due to a difference in the average orientation of line segments	14
2.5 Alternating rows of Us and inverted Us form black bars as “emergent” features	15
2.6 Natural textures demonstrating Rao’s taxonomy	16
2.7 A preattentively segmentable image. Regions differ in dot density	19
2.8 An image not preattentively segmentable. Columns of words and nonsense words	19
2.9 Texture pair consisting of randomly “thrown” micropatterns	21
2.10 Texture pair whose texels differ in the number of textons (terminations)	23
2.11 Texture pair whose texels have the same number of textons	23
2.12 Region discrimination due to configurational differences	24
2.13 Images consisting of three regions	26

	x
2.14 Treisman's feature integration model	28
2.15 Overlapping texture pair	29
2.16 Typical filter-bank architecture	34
3.1 Example of the real part of a GEF	38
3.2 Schematic of the frequency domain representation of a GEF	39
3.3 Texture sequence demonstrating the need for a nonlinearity	41
4.1 Examples of two levels of texture complexity: (a) uniform and (b) nonuniform	45
5.1 Bipartite textured image model	60
5.2 Example Gabor-filter output from a nonuniformly textured image	76
6.1 Schematic representation of the application of a Gabor filter to a uniformly textured image	83
6.2 Example of a “rosette” pattern of bandpass filters	91
7.1 Contrasting outputs from C1 and C2 configuration filters	106
8.1 Examples of 1-D textures	115
8.2 1-D Gabor-filter outputs m derived from 1-D textures	116
8.3 Uniformly textured image consisting of +s and Ls	121
8.4 Uniformly textured image consisting of texels that differ in orientation	125
8.5 Uniformly textured image with a different texel spacing in the x and y directions	127

8.6 Uniformly textured image with regions shifted both horizontally and vertically, thus producing a texture-phase discontinuity	130
8.7 Uniformly textured image with each textured region having a different texel spacing	133
8.8 Nonuniformly textured image consisting of +s and Ls	136
8.9 Difference-in-variation signature resulting from arrows-and-triangles image	138
8.10 Natural texture pair consisting of Brodatz's "grass lawn" (D9) and "cotton canvas" (D77)	142
8.11 Natural texture pair consisting of Brodatz's "straw matting" (D55) and "raffia" (D84)	144
8.12 Natural texture pair consisting of Brodatz's "pressed cork" (D4) and "beach sand" (D29)	147
8.13 Beck's emergent features	150

Chapter 1

Thesis Overview

Texture segmentation, which is the partitioning of an image into homogeneous textured regions, continues to be a challenging problem in computer vision. Classic statistical and structural approaches, while applicable to many computer vision problems, typically focus on particular image attributes that characterize the textures of interest. Hence, the variety of textures that can be successfully segmented is limited. The human visual system, on the other hand, can segment textures robustly. This realization has motivated researchers in the fields of computer vision, psychophysics, and neurophysiology to study how humans perceive textures and has resulted in a promising new approach to texture analysis.

This new approach is based on the concept of local spatial frequency. Unlike classical Fourier analysis, where frequency refers to sinusoids of infinite extent, the new approach views frequency as a local phenomena (a local frequency) that can vary with position throughout an image. Textures are characterized by their local spatial frequency content. Two textures, then, can be segmented based on local-frequency differences.

One popular method for extracting these local frequencies is to apply a bank of bandpass filters to an image. This results in a collection of subimages, where each subimage contains a limited range of local spatial frequencies. Motivation for this approach comes partly from psychophysical and neurophysiological evidence suggesting that the human visual system might be performing this function. Although several filter-bank

algorithms have shown promising results and some analysis has been done previously, the relationship between texture differences and the filter configurations required to discriminate those differences remains unknown. Specifying an appropriate filter configuration involves two parts: (1) designing individual filters and (2) specifying filter interactions. This thesis addresses the design of individual filters (portions of this work also appear in [1, 2, 3, 4, 5]).

The goal is to design filters that map textural differences to a difference in average filter output so that simple discontinuity detectors (e.g., an edge detector) can be used to segment a textured image. By analyzing filter output characteristics as a function of filter parameters and textural differences, suitable mappings have been found for a wide range of textures. Details of the filter used in this study can be found in Chapter 3.

Analysis based on a 1-D texture model shows that applying properly configured bandpass filters to textured images produces distinct discontinuities at texture boundaries (Chapter 4). Depending on the nature of the texture difference, these discontinuities exhibit one of two characteristic signatures: a step (Fig. 1.1a) or a valley (Fig. 1.1b). Experimental evidence indicates that these signatures are useful for texture segmentation.

Additional insight is provided in Chapter 5 by extending the analysis to 2-D. In 2-D, texture is modeled as collection of primitive geometric objects called *texels*. A homogeneous textured region consists of similar texels, and texture differences are induced by varying the type and/or organization of the texels. For convenience two levels of textural complexity are recognized: uniform and nonuniform. Uniformly textured regions consist of identical texels arranged in a regular lattice. For nonuniformly textured regions, the

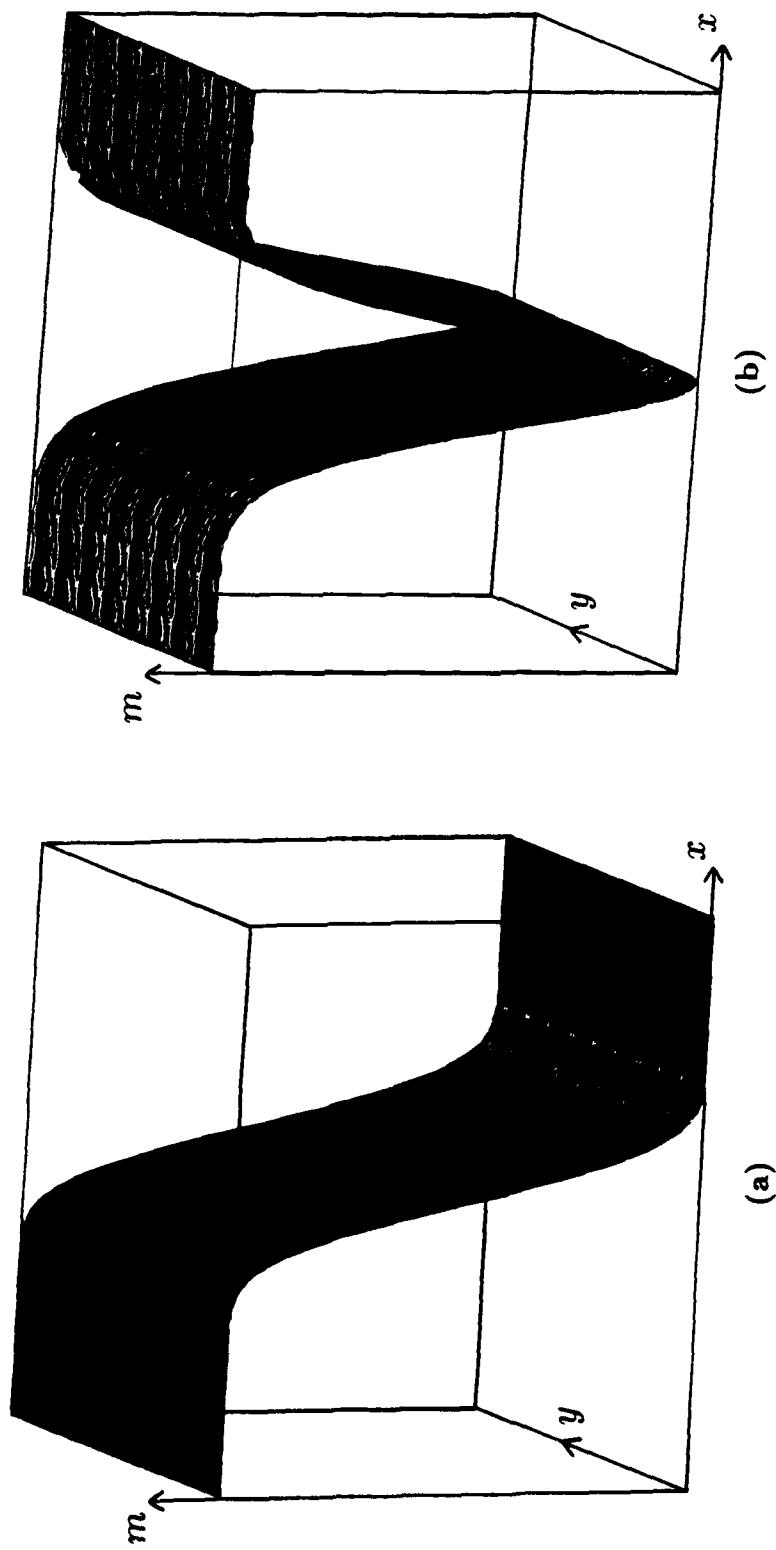


Fig. 1.1. Examples of two characteristic signature types: (a) step and (b) valley.

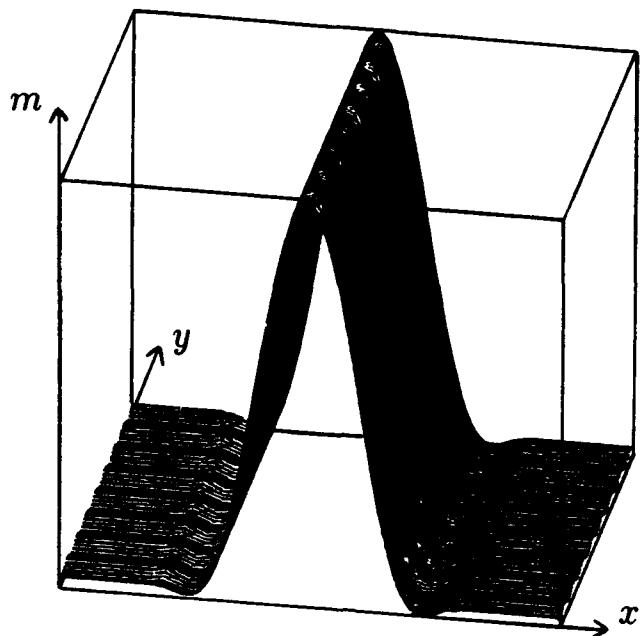


Fig. 1.2. Example of the ridge signature.

texels may vary in orientation and their shape and positions may be perturbed.

In addition to the step and valley signatures predicted by the 1-D model, the 2-D model shows that a ridge signature can also occur (Fig. 1.2). Analysis based on uniform textures shows that the step signature occurs when two textured regions differ in constituent texels (Fig. 1.3a). On the other hand, the valley and ridge signatures occur when two regions exhibit a texture-phase difference (Fig. 1.3b), resulting from spatial shifts between regions. The analysis also provides specific guidelines for selecting filter parameters to produce quality signatures (Chapter 6). In particular, the conditions favoring asymmetric filters are revealed—an issue not previously addressed.

For nonuniform textures, a detailed analysis is impractical due to their complexity. Experimental results suggest, however, that the signatures found for uniform textures

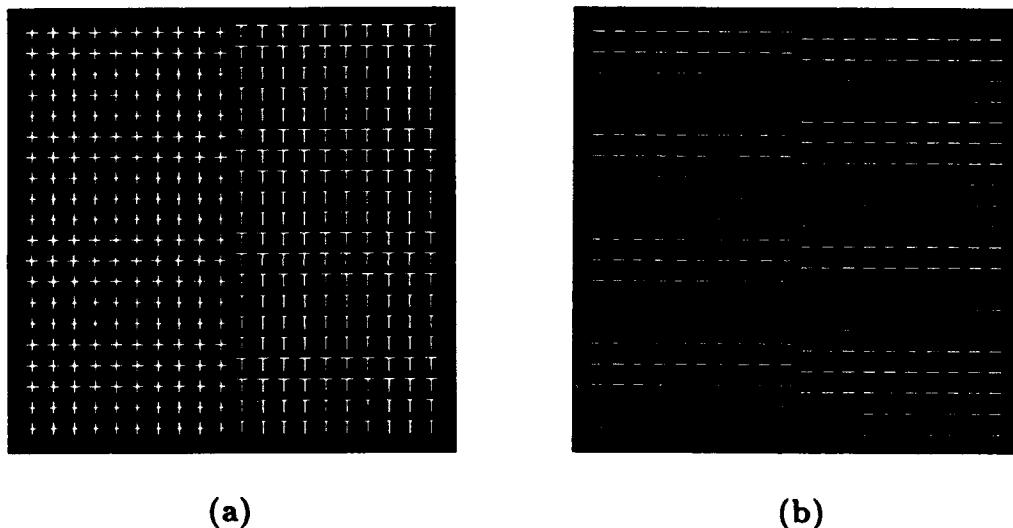


Fig. 1.3. Textured regions exhibiting two types of texture difference:
(a) Texel difference—regions differ in constituent texels.
(b) Texture-phase difference—spatial shift between regions.

occur for textures in general. Due to the texel variation in nonuniform textures, though, these signatures can exhibit local output variations, which can hinder segmentation. By judicious selection of filter parameters coupled with post-filter smoothing, distinct signatures can often be achieved, and the image can be easily segmented. Unfortunately, the guidelines for selecting filter parameters, which were developed analytically for uniform textures, are only approximately correct for nonuniform textures. To overcome this problem, an algorithm is developed in Chapter 7 to find the “best” filter parameters for any given texture pair. The algorithm has been applied successfully to a variety of textures including synthetic, natural, uniform, and nonuniform.

In addition to the three signature types mentioned earlier, texel variation in nonuniform textures can produce a fourth signature type, which is a step change in average local output variation (Fig. 1.4). Although this signature does not conform to the design goal mentioned previously, simple post-filtering operations can transform this signature into a step signature.

Chapter 8 presents experimental evidence supporting the analyses. Examples are provided demonstrating the signatures mentioned above and the texture/filter combinations that produced them. Chapter 9 provides concluding remarks.

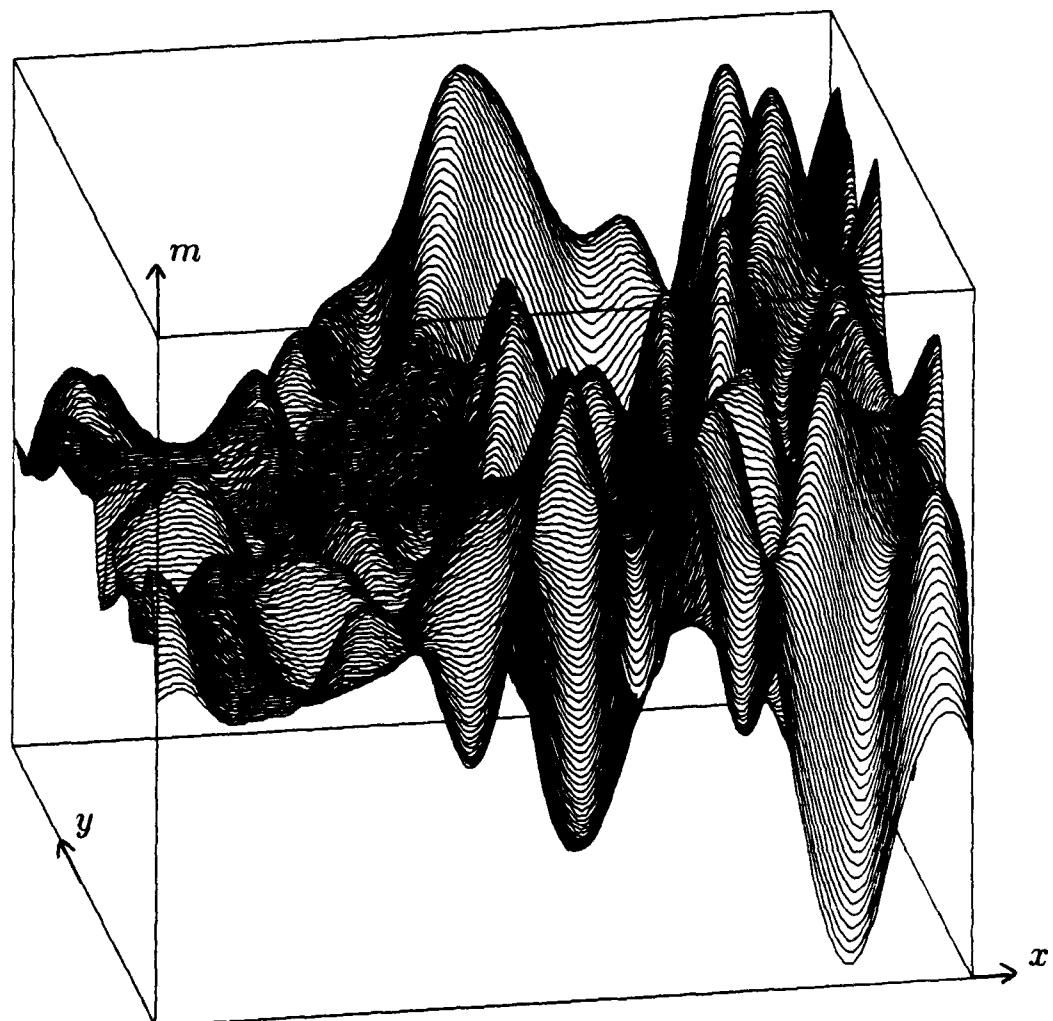


Fig. 1.4. Example of signature exhibiting a step change in average local output variation.

Chapter 2

Introduction

This thesis concerns computational methods for analyzing texture. But, what exactly is texture? The dictionary describes texture as “the visual and especially tactile quality of a surface” [6]. Some examples that come to mind are a grass lawn, a sandy beach, and a woven fabric. The dictionary goes on to characterize texture as the “...physical structure given to an object by the size, shape, arrangement, and proportions of its parts” [6]. Referring to the previous examples, this structure is formed by the blades of grass, the particles of sand, and the weave in the fabric. Although textured surfaces are inherently three dimensional, the emphasis of my research is on monocular vision. Thus, subsequent discussion and analyses are limited to the planar projections of textures called textural images. Also, the structural properties of texture are of primary interest; so, color and average intensity differences between textures are ignored.

The analysis of textured images can be divided into four categories: discrimination, segmentation, classification, and shape from texture. In texture discrimination, the goal is to determine whether or not a texture difference exists between two regions of an image. Referring to Fig. 2.1, for example, the task might be to determine if region I differs from region II. Texture segmentation involves partitioning an image into regions of homogeneous texture. Three such partitions are shown in Fig. 2.1. The difference between segmentation and discrimination is that segmentation determines the boundaries between textured regions, whereas in discrimination, the regions are known *a priori*;

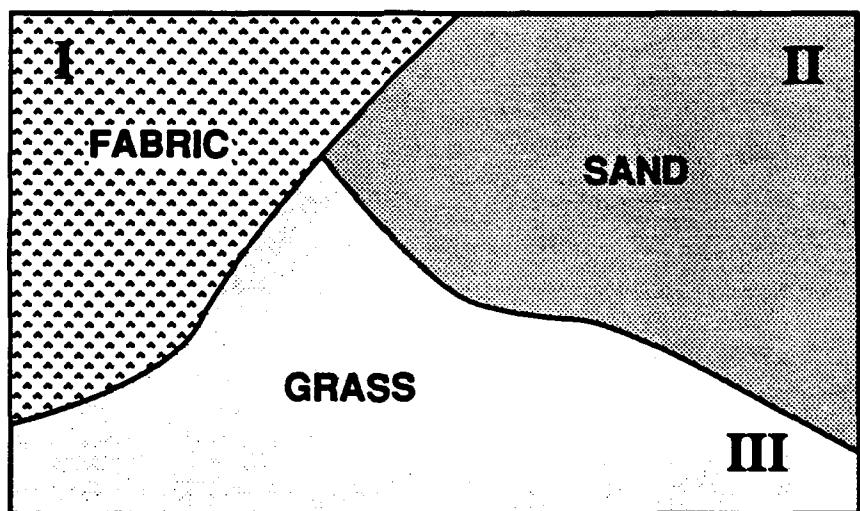


Fig. 2.1. Schematic representation of three differently textured regions.

thus, segmentation is more difficult. Once an image is properly segmented into regions, texture classification can be used to identify each region by type; e.g., as grass, sand, or fabric. Texture analysis can also provide clues to the shape of objects. A simple example of shape from texture is the use of texture gradients for determining surface orientation [7]. For example, a surface that is oblique to the viewing plane produces image structure that changes scale with image coordinates (see Fig. 2.2). Measuring this change in scale can provide an estimate of surface orientation. As with classification, an image must first be segmented before texture gradients can be computed. My research is primarily concerned with techniques for texture discrimination and segmentation. And, though the methods developed in this thesis can be extended to solve classification problems, the study of texture classification and shape from texture is beyond the scope of this work.

A major problem in developing robust methods for texture analysis is the lack of a precise definition for texture. Although an intuitive description of texture was given earlier, it is far from a comprehensive definition. Texture, it seems, is one of those terms that defies mathematical definition.

To illustrate the difficulty in defining texture, consider how humans segment the following textures. (These examples, which are commonly referred to as synthetic textures, typify those developed by researchers to exhibit specific textural properties [8, 9, 10, 11].) Fig. 2.3a consists of a region of Rs and another of mirror-image Rs. Note that, although it is easy to distinguish between a single R and its mirror image, considerable effort is required to determine the region boundary in this figure. Evidently, the random orientations mask the differences between texels. This is not always the case, as

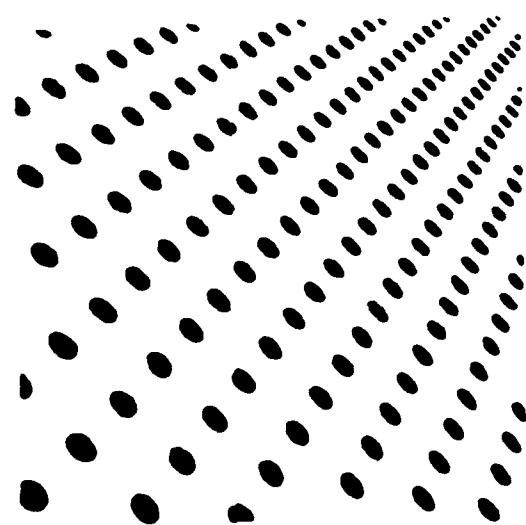


Fig. 2.2. Surface orientation perceived due to a texture gradient (from Blostein and Ahuja [7]).

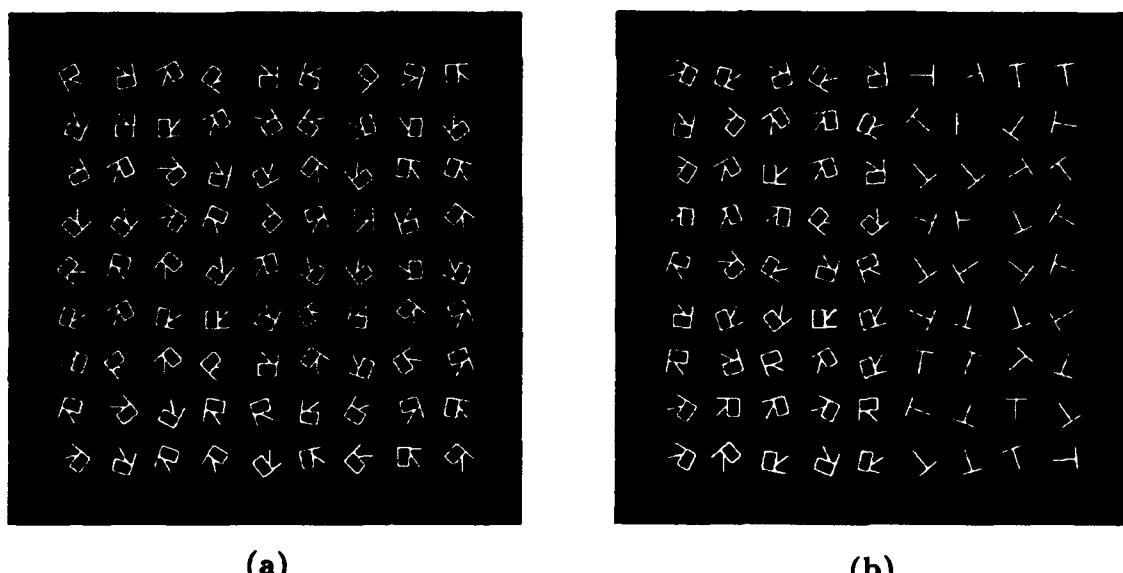


Fig. 2.3. Textures with randomly oriented texels:

- (a) **Pair of textures consisting of Rs on the left and mirror-image Rs on the right. Texture pair not easily distinguishable.**
- (b) **Pair of textures consisting of Rs on the left and Ts on the right. Texture pair easily distinguishable.**

demonstrated in Fig. 2.3b. Here, the Rs are easily distinguished from the Ts in isolation or when randomly oriented in a texture. In many cases, differences in orientation can cause segmentation. For example, the center region in Fig. 2.4 is easily distinguishable from the background even though the lines in the two regions differ only slightly in average orientation [8]. Finally, consider Fig. 2.5. The bottom region consists of alternating columns of Us and inverted Us, whereas the top consists of alternating rows. For most observers, however, it is the 5 horizontal black bars (so called "emergent" features) that seem to attract attention [9]. Note that the actual boundary between regions goes unnoticed. Thus, in this case, the difference in texture is not even the dominant feature. It is important to point out that using human performance as an indicator of textural differences is not the only alternative. For instance, a particular application might require distinguishing Rs from mirror-image Rs. Thus, what constitutes a textural difference can depend on the application. As these examples demonstrate, textural differences can be difficult to characterize.

In the absence of a precise definition for texture, researchers have resorted to more qualitative descriptions. Rao has proposed that textures can be grouped into four classes: strongly ordered, weakly ordered, disordered, and compositional [12]. Fig. 2.6 shows examples of naturally occurring textures (from Brodatz [13]) illustrating Rao's taxonomy. Fig. 2.6a is "cotton canvas"—an example of a strongly ordered texture. This class is characterized by an arrangement of primitive geometric shapes called texels. Fig. 2.6b is "straw". In this example, the characteristic feature is a globally oriented structure. Textures with this property are called weakly ordered. The third example (Fig. 2.6c) is "grass lawn"—a disordered texture. This class exhibits no obvious pattern

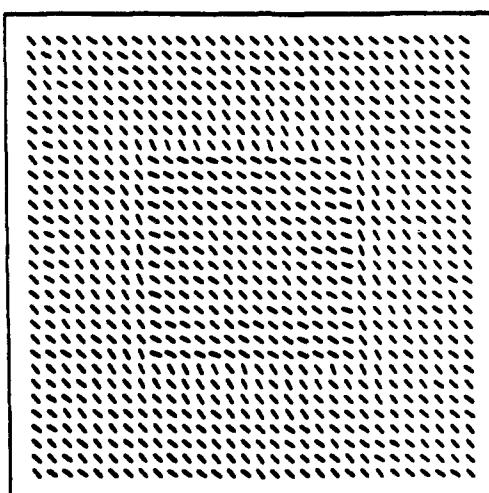


Fig. 2.4. Segmentation due to a difference in the average orientation of line segments (from Nothdurft [8]).

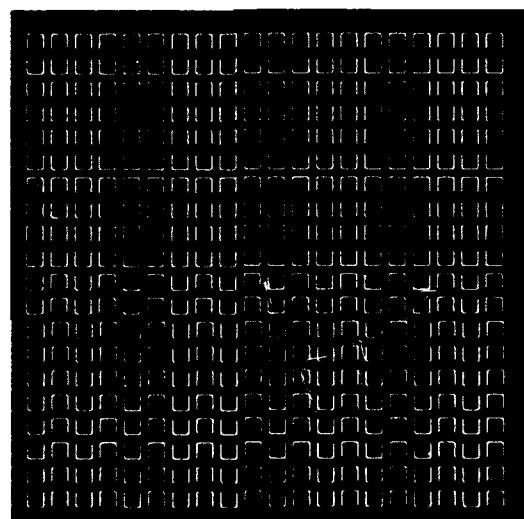


Fig. 2.5. Alternating rows of Us and inverted Us form black bars as “emergent” features. Boundary between regions of alternating rows and alternating columns is not the dominant feature (from Beck *et al.* [9]).

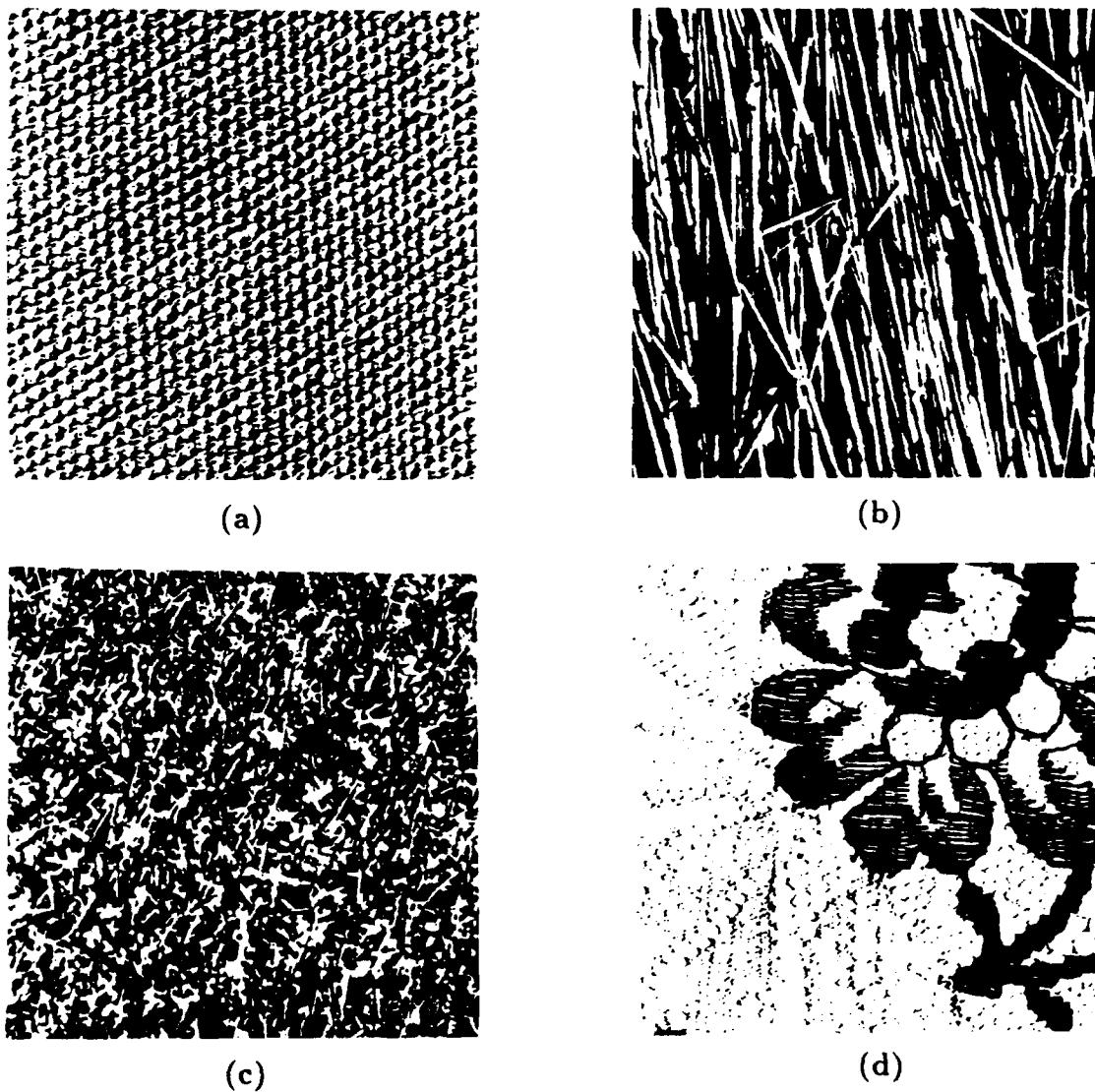


Fig. 2.6. Natural textures demonstrating Rao's taxonomy (from Brodatz [13]):
(a) "cotton canvas"—strongly ordered;
(b) "straw"—weakly ordered;
(c) "grass lawn"—disordered;
(d) "lace"—compositional.

of texels nor a dominant orientation, rather the texture seems to be described best by the statistical distribution of image pixels. Compositional textures, such as the example in Fig. 2.6d, are just combinations of the other three classes.

Classic texture-analysis techniques tend to be divided along these class boundaries. For example, statistical methods [14, 15, 16, 17], random field models [10, 14, 18, 19, 20], and fractals [21, 22] have been used to model disordered textures, while collections of geometric primitives [23, 24, 25] and mosaic models [26] have been used for strongly-ordered textures. Rao recommends modeling weakly-ordered textures using orientation fields [12]. Little has been done with compositional textures.

While these methods can be effective for textures within a particular class, performance is typically poor outside the class. The human visual system, on the other hand, can analyze textures robustly. This realization has motivated researchers in the fields of computer vision, psychophysics, and neurophysiology to study how humans perceive textures in an effort to develop more robust machine-vision texture-analysis schemes.

Early insights into human texture-perception mechanisms were provided by psychophysical experiments [10]. These experiments test human response to carefully controlled stimuli. By controlling stimulus properties, researchers attempt to deprive the visual system of familiar cues and to force it to rely on primitive mechanisms. Experiments such as these have shown that texture segmentation is a spontaneous process not involving conscious comparisons, suggesting that texture perception differs from form perception [27, 28]. To illustrate this point, consider the examples in Figs. 2.7 and 2.8. In Fig. 2.7, it is easy to recognize the two different regions; however, in Fig. 2.8, considerable effort is required to discover that the three columns on the left are words, and the

three columns on the right are nonsense words [29]. These dramatic differences in visual performance prompted Neisser to propose that the human visual system consists of two operating modes, a *preattentive* mode and an *attentive* mode [30]. The preattentive mode processes a large portion of the visual field quickly (presumably in parallel) but imprecisely. In contrast, the attentive mode operates within a much smaller aperture and at a much slower rate but can perform more detailed image analysis. The rapid discrimination and segmentation of textures is attributed to the preattentive mode and is often termed preattentive texture discrimination (or segmentation). The perception of form, on the other hand, requires scrutiny, suggesting that the attentive mode is needed for this task.

The notion of a fast, parallel, imprecise system is the foundation of many modern texture-analysis models. Section 2.1 describes early efforts to model human preattentive texture discrimination. These methods attempt to characterize texture differences by the statistical properties of image pixels or by differences in geometric features such as edges, lines, and blobs. Section 2.2 introduces a more recent approach that involves detecting local spatial-frequency differences between textured regions. It also describes the application of the local spatial-frequency approach to texture analysis and discusses open questions leading to this thesis.

2.1 Early Texture Perception Models

The first attempts to model human texture perception were made by Julesz in the early 1960's [10]. He set out to determine if texture discrimination could be described by the statistical properties of textures alone, or whether it was necessary to consider

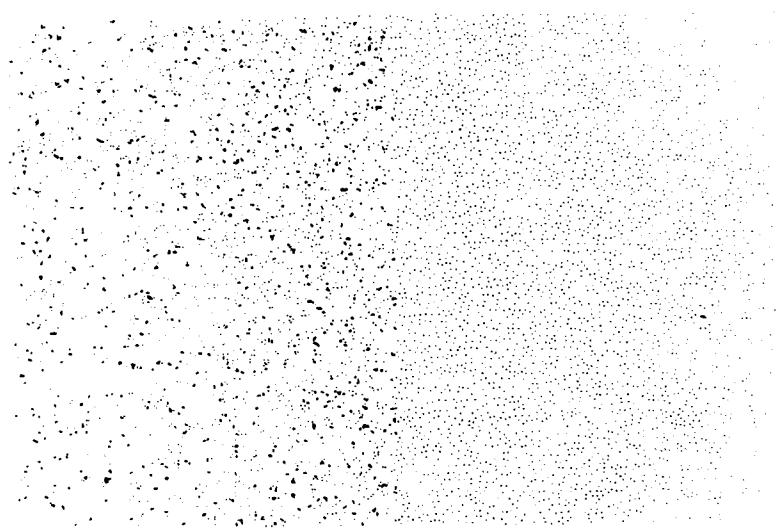


Fig. 2.7. A preattentively segmentable image. Regions differ in dot density (from Julesz *et al.* [29]).

METHODS SCIENCE COLUMNS NIATREC YLKCIUQ DEMCNUP
SUBJECT MERCURY GOVERNS ECWEICS YFICEPS ESICERP
COLUMNS CERTAIN QUICKLY SDONTEN SDROCR EZIDIXO
RECORDS EXAMPLE SCIENCE STCIPED HSILONE NIATREC
MERCURY SPECIFY PRECISE TCEJBUS DEMCNUP SHREVOG
METHODS COLUMNS MERCURY ELPMAKE YLKCIUQ YFICEPS
CERTAIN DEPICTS ENGLISH ECWEICS ESICERP ELPMAKE
QUICKLY METHODS EXAMPLE YFICEPS YRUCREM DEMCNUP
ENGLISH SUBJECT RECORDS ELPMAKE SHREVOG EZIDIXO
MERCURY PUNCHED CERTAIN SNMULCC TCEJBUS ESICERP
SUBJECT OXIDIZE GOVERNS HSILONE SDROCR ELPMAKE
METHODS ENGLISH COLUMNS NIATREC ESICERP DEMCNUP
DEPICTS SPECIFY PRECISE EZIDIXO YLKCIUQ ECWEICS
CERTAIN RECORDS SCIENCE STCIPED ELPMAKE HSILONE
PRECISE QUICKLY METHODS YFICEPS YRUCREM SHREVOG

Fig. 2.8. An image not preattentively segmentable. Columns of words and nonsense words (from Julesz *et al.* [29]).

the internal operation of the visual system. He proceeded to generate binary (black and white) texture pairs whose pixels had predetermined n^{th} -order joint probability distributions.¹ Each textured region consisted of a collection of micropatterns, called texture elements (texels), either regularly spaced or thrown at random (Fig. 2.9). Julesz observed that most textures that have different first or second-order distributions are easily discriminated, whereas textures that differ only in higher order distributions are not discriminable. Counterexamples, however, were found [18, 29], which lead Gagaliowicz to propose an alternative explanation. He observed that due to inhomogeneities in constructing stochastic textures, local statistics differed greatly from the global statistics [18]. He proposed that humans discriminate textures based on local computations, and that humans cannot discriminate textures that have the same local second-order statistics.

At about the same time, Julesz proposed a more localized model with his texton theory [23, 24]. He found that many textures that have different first or second-order distributions also differ in some local features, which Julesz called *textons*. He proposed that preattentive texture discrimination is due either to differences in texton type or to differences in the number of textons. Features considered to be textons included color, elongated blobs (with some orientation, width, and length), terminations, and crossings (i.e., points where lines intersect) [31]. Note that under this theory, the relative positions of textons is unimportant for texture discrimination - only the number or type is significant. Therefore, configurational differences are explicitly ignored. Fig. 2.10 is

¹A first-order probability distribution refers to a probability distribution with one random variable, a second-order joint distribution refers to a probability distribution involving two random variables (pairs of pixels), etc.

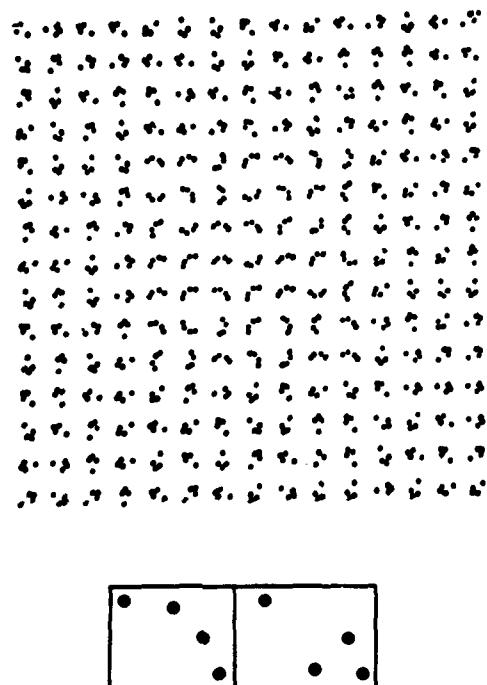


Fig. 2.9. Texture pair consisting of randomly “thrown” micropatterns. Regions preattentively discriminable (from Julesz [24]).

a texture considered to be discriminable due to a difference in the number of textons (terminations) [31]. Discrimination fails, though, in Fig. 2.11 due to a lack of any texton difference [31].

Texton theory, however, has certain limitations. First, textons are defined only verbally (e.g., elongated blob), and thus their characteristics must be inferred from examples [11, 32]. Since textons are only vaguely defined, it seems one must solve the pattern-recognition problem to recognize a texton. Second, there is some doubt that terminators and crossings are textons [11, 28, 33]. Third, the rejection of possible configurational effects seems inconsistent with the results of others [9, 34].

An alternative to texton theory was proposed by Beck *et al.* [9]. They observed that texture discrimination occurs easily when textured regions differ in the slopes, sizes, colors, and brightness of the texels or their component parts. They also observed that differences in texel configuration can affect discrimination (Fig. 2.12). Based on these observations, they proposed (similar to Julesz's texton theory) that texture discrimination is based on first-order differences (i.e., differences in distribution) in image features. However, unlike Julesz, they suggested that these features are not computed directly from the retinal array. Instead, only a few simple features are detected directly. These features are then “linked” into higher order texels based on the Gestalt heuristics of proximity, similarity, and good continuation [35]. Texture discrimination is the result of feature differences between texels. Beck *et al.* do not define their primitive features explicitly. Rather, they describe them as those objects that best stimulate the simple cells in the visual cortex (e.g., the edges and bars proposed by Hubel and Wiesel [36]).

↙ ↘ ↗ ↙ ↙ ↙ ↙ ↘ ↘ ↘
 ↙ ↙ ↘ ↙ ↙ ↙ ↙ ↙ ↘ ↘
 ↘ ↗ ↗ ↙ ↙ ↙ ↙ ↙ ↘ ↘
 ↗ ↙ ↙ ↙ ↙ ↙ ↙ ↙ ↘ ↘
 ↙ ↘ ↙ ↙ ↙ ↙ ↙ ↙ ↘ ↘
 ↙ ↘ ↙ ↙ ↙ ↙ ↙ ↙ ↘ ↘
 ↗ ↗ ↗ ↙ ↙ ↙ ↙ ↙ ↘ ↘
 ↘ ↗ ↗ ↙ ↙ ↙ ↙ ↙ ↘ ↘
 ↗ ↙ ↙ ↙ ↙ ↙ ↙ ↙ ↘ ↘
 ↘ ↗ ↗ ↙ ↙ ↙ ↙ ↙ ↘ ↘
 ↙ ↘ ↙ ↙ ↙ ↙ ↙ ↙ ↘ ↘

Fig. 2.10. Texture pair whose texels differ in the number of textons (terminations). Regions preattentively discriminable (from Julesz [31]).

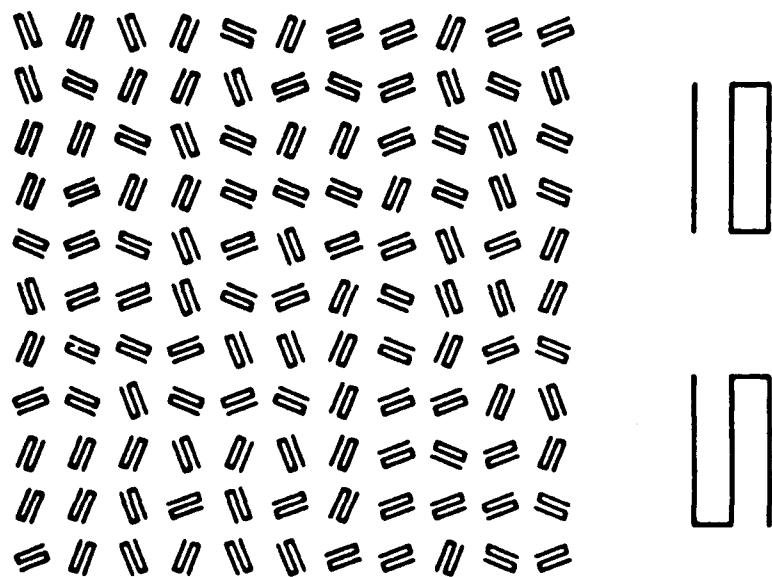


Fig. 2.11. Texture pair whose texels have the same number of textons. Regions not preattentively discriminable (from Julesz [31]).

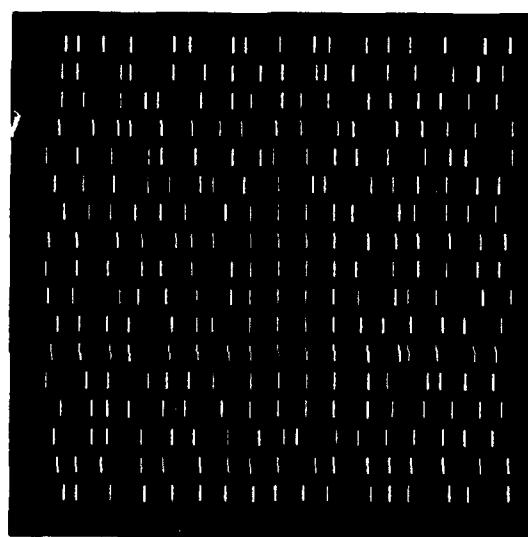


Fig. 2.12. Region discrimination due to configurational differences. Center region consists of colinear line segments (from Beck *et al.* [9]).

Similar to the linking idea of Beck *et al.*, Marr and Hildreth suggested that complex objects are formed by hierarchical grouping, starting with primitive features [37]. Marr and Hildreth, however, proposed a different set of primitive features. They suggested that the function of simple cells is to locate intensity gradients rather than edges or bars. They introduced the Laplacian-of-Gaussian operator ($\nabla^2 G$), which approximates the receptive field profiles of simple cells, and when convolved with an image, indicates the location of intensity gradients over a selected range of scales [25, 37]. The effect of the Gaussian is to lowpass filter the image, thus, eliminating fine-grain intensity changes. Applying the Laplacian to this filtered image results in zero values at the location of maximum intensity gradients. The positions where these zero values occur (called *zero crossings*) are used to form the primitive features in the Marr-Hildreth model. These zero crossings are then grouped into abstract objects called *place tokens*. Place tokens represent edges, bars, blobs, and terminators with properties such as orientation, contrast, length, width, and position. These tokens are computed over a range of scales and can themselves be grouped to form larger and larger tokens. Differences in the properties or configurations of tokens becomes the basis for texture discrimination.

Both Beck *et al.* and Marr and Hildreth attribute texture discrimination to differences in the collective properties of higher order objects (texels or place tokens). The idea of comparing collections of properties (a kind of feature vector correlation), however, is inconsistent with certain psychophysical findings. For example, Treisman found that certain combinations of otherwise discriminable features do not produce texture discrimination [28]. By way of illustration, consider the images in Fig. 2.13. The theories of Beck *et al.* or Marr and Hildreth would form three regions in each frame. In

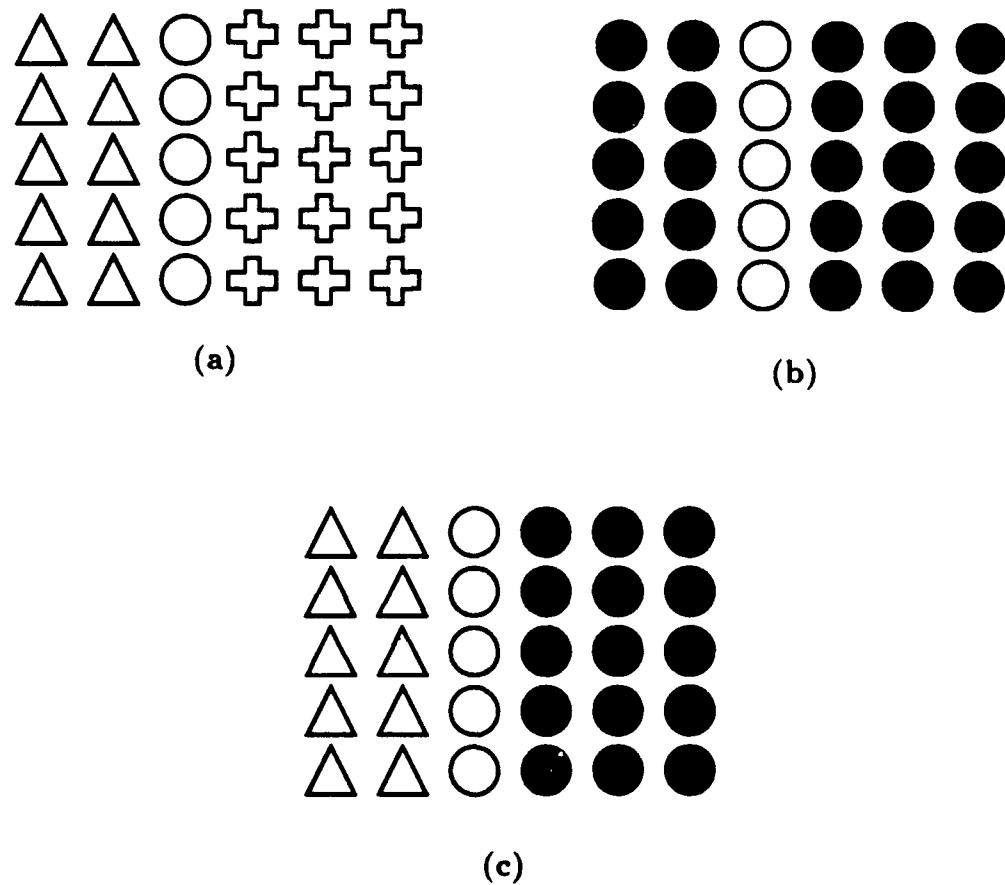


Fig. 2.13. Images consisting of three regions (from Treisman [28]). Region differences:

- (a) **Texel shape.**
- (b) **Shading.**
- (c) **Conjunction of features.**

Fig. 2.13a discrimination would be based on differences in texel shape. In Fig. 2.13b, the difference is due to shading, and in Fig. 2.13c, the dark circles would form one region, the open circles another, and the open triangles a third; however, if Fig. 2.13c is viewed quickly, only two regions are perceived. From this and other psychophysical experiments, Treisman concluded that the conjunction of features (e.g., a red, vertical, blob at position (x, y)) is not available to the preattentive system. Rather each feature has its own separate representation in space (a feature map). In this way, the position of something red can be detected, and the position of a blob can be detected, but the correspondence between red and blob is not represented directly. Only by attentive search can this correspondence be resolved. See Fig. 2.14 for a diagram of this model.

Iwama and Maida, using edge segments and terminations as primitive features, have developed a texture-segmentation architecture that combines the token idea of Marr and the concept of feature integration [38]. Their model performs well over a range of textures, and in particular, has the unusual property of being able to represent overlapping textures (Fig. 2.15). A careful examination of the implementation details, however, reveals the following limitations. First, selecting geometric objects as the primitive feature set makes it difficult to develop robust feature detectors (primarily because the features are not defined mathematically). Second, the criteria for grouping is largely unspecified. This requires the specification of many empirical parameters, making it difficult to predict performance on untested images. These problems are not unique to Iwama and Maida, but are inherent to models based on semantic descriptions.

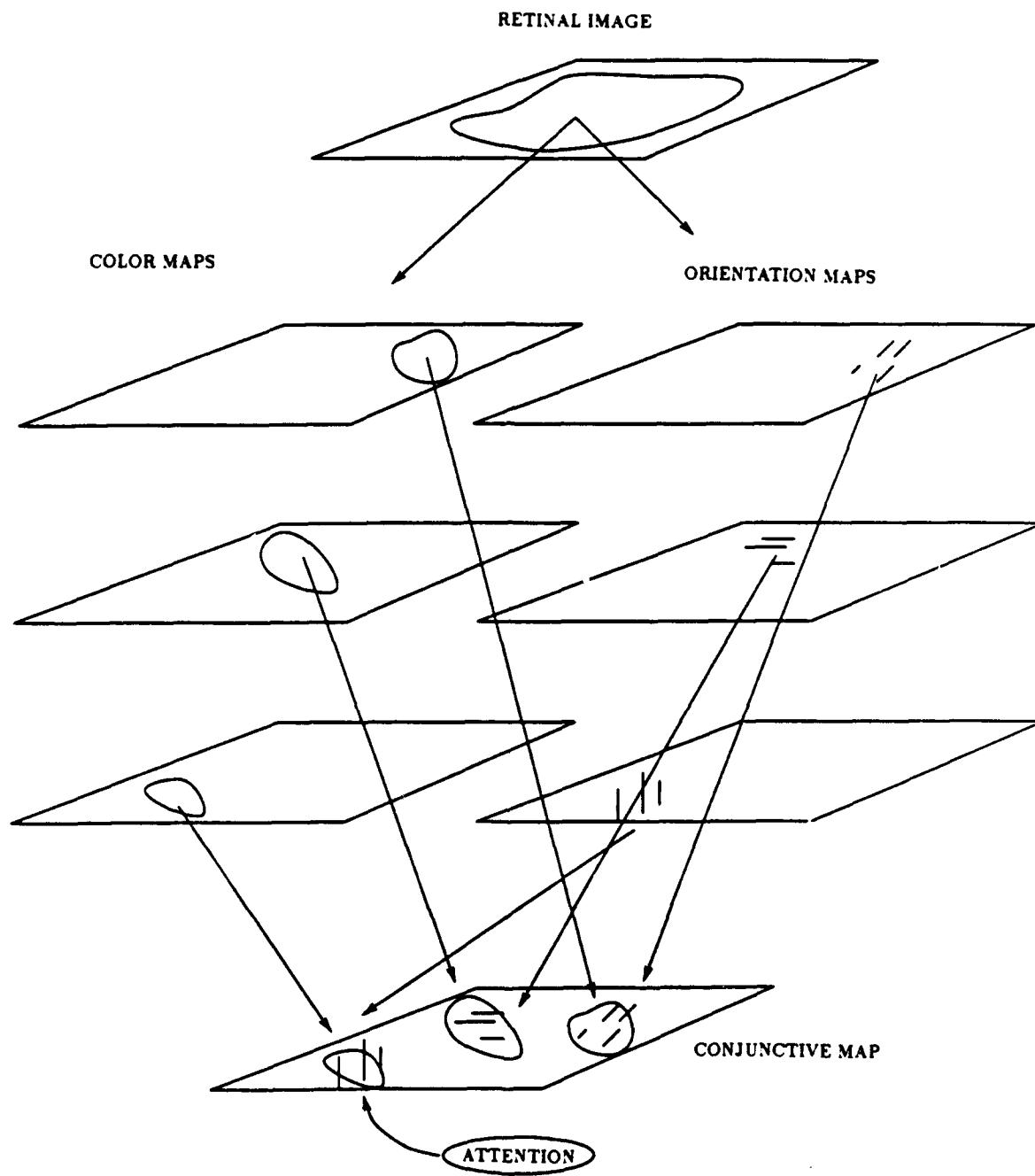


Fig. 2.14. Treisman's feature integration model (from Treisman [28]).

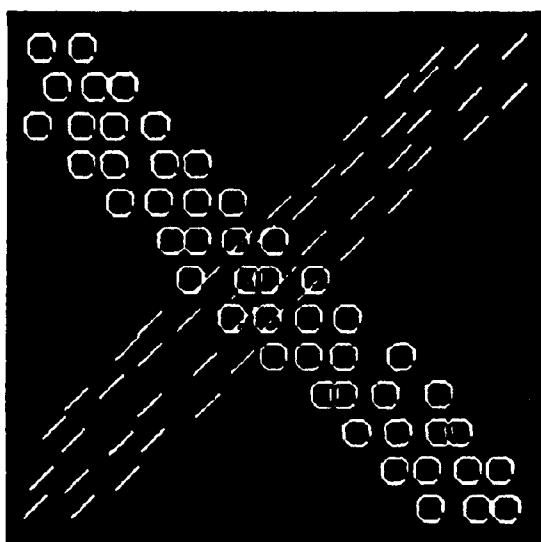


Fig. 2.15. Overlapping texture pair (from Iwama *et al.* [38]).

2.2 The Local Spatial-Frequency Approach

With the exception of Marr, the texture-perception theories mentioned thus far consider the primitive features to be geometric objects such as edges and bars. This idea was motivated largely by the findings of Hubel and Wiesel, who found that simple cells in the visual cortex responded better to these objects than to diffuse light [36]. More recently, however, an alternate interpretation of cortical cell function was proposed - that of a spatial-frequency analyzer [39, 40, 41, 42, 43, 44, 45, 46].

While measuring human contrast sensitivity to simple functions (sine waves, square waves, etc.), Campbell and Robson discovered that the response to these functions could be predicted from the frequency components of the waveform under test [39]. They proposed that the visual system behaves as a number of independent detector mechanisms each preceded by a narrow-band filter tuned to a different frequency. They suggested that each filter/detector pair constituted a separate *channel*, and each channel would have its own contrast sensitivity function (i.e., bandpass characteristics). Subsequently, neurophysiological evidence appeared suggesting that simple cells respond better to the frequency components of a stimulus than to its geometric features [40, 41]. The initial tendency was to propose that the visual system was computing the Fourier transform of the image. This idea was dispelled by Julesz and Caelli who pointed out that the Fourier transform is a global operation, and thus is not capable of representing local intensity variations explicitly [47].

The abundance of apparently conflicting evidence prompted a debate as to the functionality of simple cells: Are they feature detectors or spatial-frequency analyzers

[48]? Viewing simple cells as feature detectors implies that the representation of an image in the visual cortex is in the spatial domain, whereas viewing them as frequency analyzers suggests a spatial-frequency domain representation. Experimental work by Maffei *et al.*, however, suggests that the two views are not necessarily contradictory [45]. Their results indicate that both interpretations of simple cell function can be phenomenologically correct. This apparent paradox can be explained using the concept of local frequency.

The concept of local frequency was developed by Gabor [49] for 1-D signals and extended to images by Daugman [50]. Their work shows that the spatial representation and the spatial-frequency representation are just opposite extremes of a continuum of possible joint space/spatial-frequency representations. In a joint space/spatial-frequency representation, frequency can be viewed as a local phenomena (i.e., a local frequency) that can vary with position throughout the image. These local frequencies arise due to local interactions among groups of sinusoids. These sinusoids, which differ in frequency and phase, constructively interfere with each other to produce spatially localized concentrations of signal energy. It is this localized signal energy that forms the intensity patterns in an image.

Marcelja demonstrated the plausibility of a joint space/spatial-frequency representation in the human visual system by showing that the functions described by Gabor closely approximate the receptive field profiles of simple cells. Subsequently, additional neurophysiological support appeared [42, 44, 51, 52].

The evidence suggesting that early visual mechanisms are performing local spatial-frequency analysis spawned a number of texture-discrimination (and segmentation) models [53, 54, 55, 56, 57, 58, 59, 60]. One characteristic that distinguishes these models from feature-based models is that textural differences are not viewed as resulting from geometric feature differences. Rather, local spatial-frequency models assume that perceptually significant textural differences correspond to differences in local spatial-frequency content. By decomposing an image into a joint space/spatial-frequency representation, the distribution of local spatial frequencies within the image can be determined. This distribution of frequencies, then, becomes the basis for texture analysis.

The representation of an image in either the spatial domain or the spatial-frequency domain is unique. For any image there is only one pixel array or one Fourier transform. In the joint space/spatial-frequency domain, however, an infinite number of representations are possible, and several techniques are available for performing the decomposition. One method for decomposing an image into a joint space/spatial-frequency representation is the windowed Fourier transform. To compute the continuous windowed Fourier transform, the following equation is evaluated (shown in 1-D for simplicity).

$$h(x, f) = \int_{-\infty}^{\infty} g(z) w^*(z - x) e^{-i2\pi f z} dz \quad (2.1)$$

Here w^* is the complex conjugate of the window function w , g is the function to be transformed, and $h(x, f)$ is the joint-domain representation of the (1-D) “image.” The special case when the window function is a Gaussian is called the Gabor transform [49, 61]. Note that the windowed Fourier transform is similar to the classic Fourier

transform except that the input is multiplied by a window function, whose position is parameterized. In effect, (2.1) computes the Fourier transform of a subset of the original image - hence the term *local frequency*.

Although (2.1) can be viewed as computing a local Fourier transform, there is another useful interpretation. Note that

$$w^*(z - x)e^{-i2\pi fz}$$

in (2.1) is a modulated window function and therefore has bandpass characteristics. Thus, (2.1) can be interpreted as filtering the image g with a bandpass filter, where the center frequency of the filter is f and its bandwidth is determined by the window function.

The application of bandpass filters to images is an integral part of many texture-analysis schemes, including those based on Gabor elementary functions [53, 55], wavelet transforms [62, 63], derivatives of Gaussians (Hermite polynomials) [64], and differences of offset Gaussians [56, 64]. Although methods differ in the bandpass characteristics of the individual filters and how the filters are distributed over the frequency domain, they can be collectively referred to as *filter-bank* models. A schematic of a typical filter-bank architecture is shown in Fig. 2.16. While the filter-bank paradigm has shown potential and some analytical work has been done to demonstrate the efficacy of certain types of filters, *the relationship between texture differences and the filter configurations required to discriminate them remains largely unknown*. Two major issues arise: (1) the design of individual filters, and (2) the configuration of the filter bank. An adequate understanding

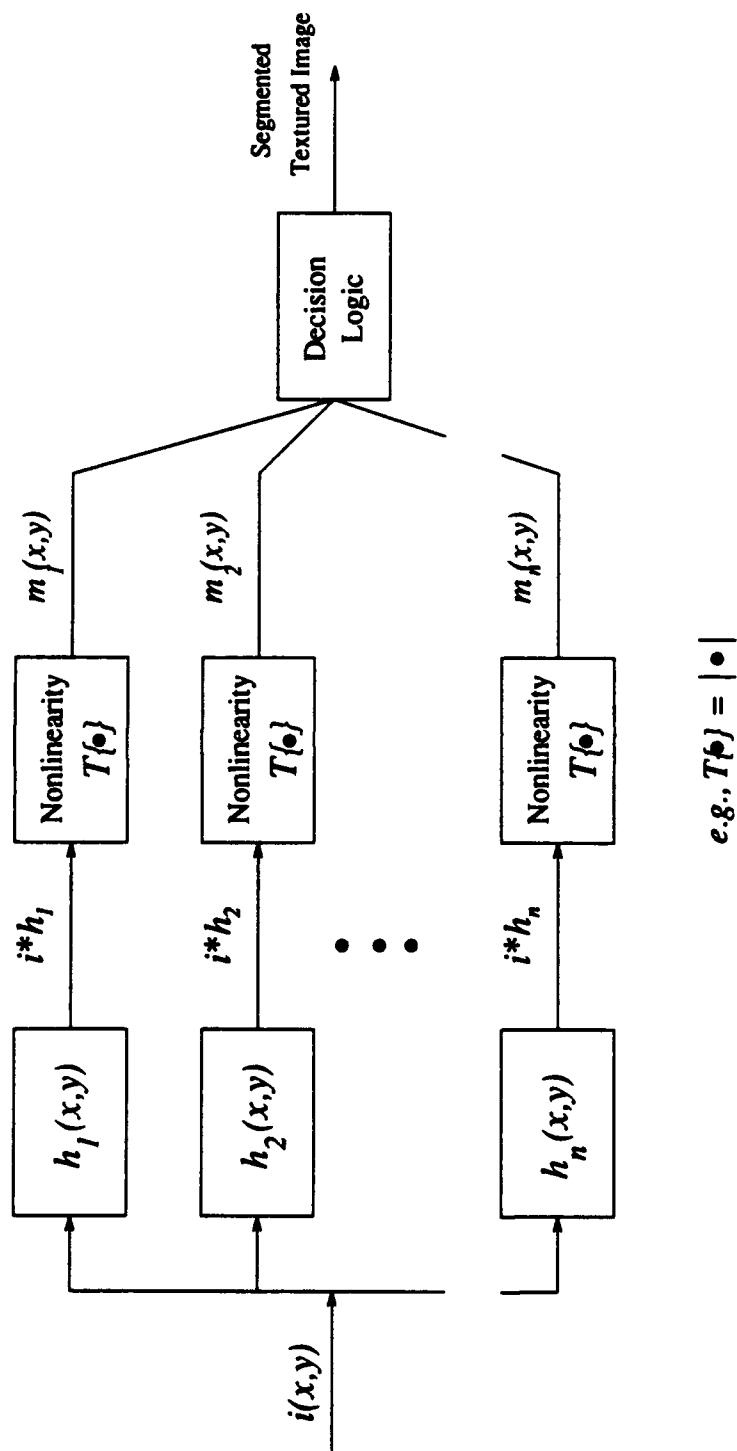


Fig. 2.16. Typical filter-bank architecture.

of how to design an individual filter seems essential for understanding how to build a suitable filter bank. Therefore my thesis addresses the issue of filter design.

The following chapters provide a detailed analysis of filter design. The analysis assumes that the filter is based on a Gabor elementary function, which is a Gaussian modulated by a complex sinusoid (i.e., it assumes a Gaussian window function in (2.1)). The analysis shows, however, that it is the bandpass characteristic of a filter function that is essential. Thus, other filter functions could conceivably be used.

Chapter 3

Defining the Filter

Subsequent analyses of textured images assumes the following filter structure:

$$m(x, y) = G_f(i(x, y)) \triangleq |i(x, y) * h(x, y)| \quad (3.1)$$

where $*$ denotes convolution, i is an image, h is a Gabor elementary function (GEF), and m is the filter output. The filtering operator G_f in (3.1) will be called a *Gabor filter*.

The form of the Gabor filter is justified below.

GEFs possess three desirable properties for texture analysis:

- The GEFs are the only functions that achieve the lower bound of the space-bandwidth product as specified by the uncertainty principle [65]. This means that they can simultaneously be optimally localized in both the spatial and spatial-frequency domains. Thus, GEFs can be designed to be highly selective in frequency, while displaying good spatial localization.
- The shapes of GEFs resemble the receptive field profiles of the simple cells in the visual pathway [44, 46].
- They are bandpass filters. Thus, GEFs can be configured to extract a specific band of frequency components from an image.

CEFs were first defined by Gabor [49] and later extended to 2-D by Daugman [50]. (A few researchers have referred to GEFs as *Gabor wavelets* [62, 63].) A GEF is a Gaussian modulated by a complex sinusoid [49, 50, 53]

$$h(x, y) = g(x', y') \exp[j(Ux + Vy)] \quad (3.2)$$

where $(x', y') = (x \cos \theta + y \sin \theta, -x \sin \theta + y \cos \theta)$ are rotated spatial-domain rectilinear coordinates, (u, v) are frequency-domain rectilinear coordinates, and (U, V) give the particular 2-D frequency of the complex sinusoid. $\phi \triangleq \tan^{-1}(V/U)$ specifies the orientation of the sinusoid, $g(x, y)$ is the 2-D Gaussian

$$g(x, y) = \frac{1}{2\pi\sigma_x\sigma_y} \exp \left\{ -\frac{1}{2} \left[\left(\frac{x}{\sigma_x} \right)^2 + \left(\frac{y}{\sigma_y} \right)^2 \right] \right\} \quad (3.3)$$

and (σ_x, σ_y) characterize the spatial extent and bandwidth of h . The aspect ratio of $g(x, y)$ is given by $\lambda \triangleq \sigma_y/\sigma_x$ and gives a measure of the filter's asymmetry. An example of the real part of a GEF is shown in Fig. 3.1. The Fourier transform of h is

$$H(u, v) = \exp \left\{ -\frac{1}{2} [(\sigma_x[u - U]')^2 + (\sigma_y[v - V]')^2] \right\} \quad (3.4)$$

where $[(u - U)', (v - V)'] = [(u - U) \cos \theta + (v - V) \sin \theta, -(u - U) \sin \theta + (v - V) \cos \theta]$ are shifted and rotated frequency coordinates. $H(u, v)$ is a Gaussian that is shifted (U, V) frequency units along the frequency axes (u, v) and rotated by an angle θ relative to the positive u axis. Thus, H acts as a bandpass filter with center frequency (U, V) [relative to (u, v)] and a bandwidth controlled by σ_x and σ_y . Note that when the aspect ratio

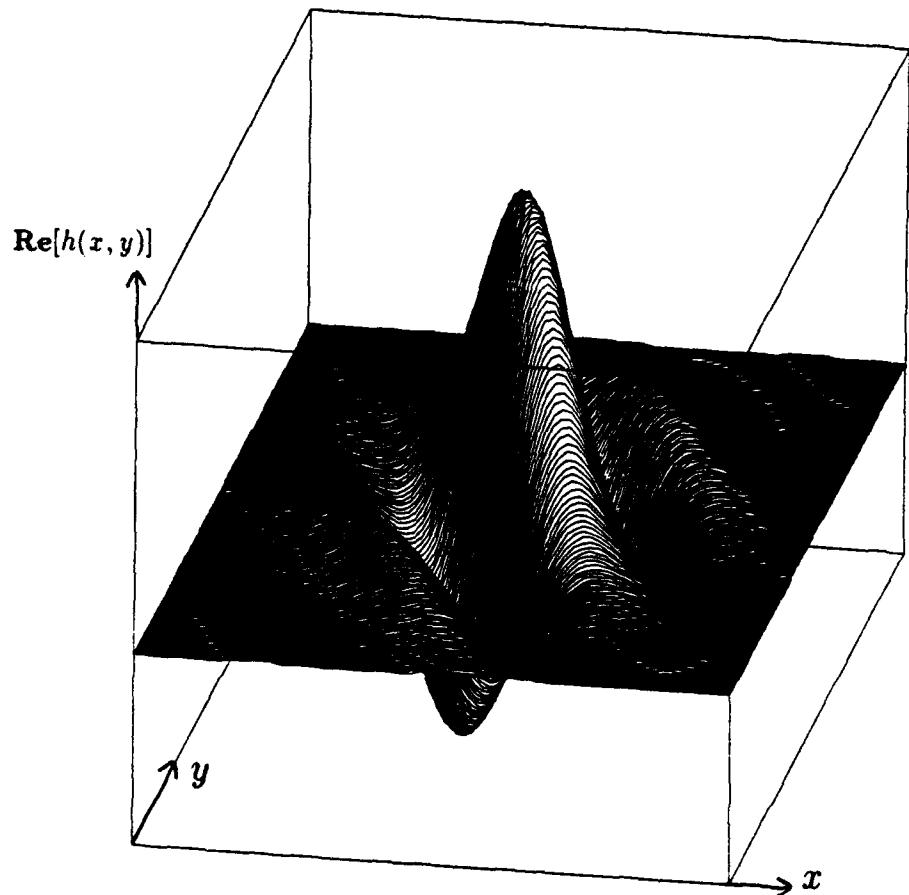


Fig. 3.1. Example of the real part of a GEF.

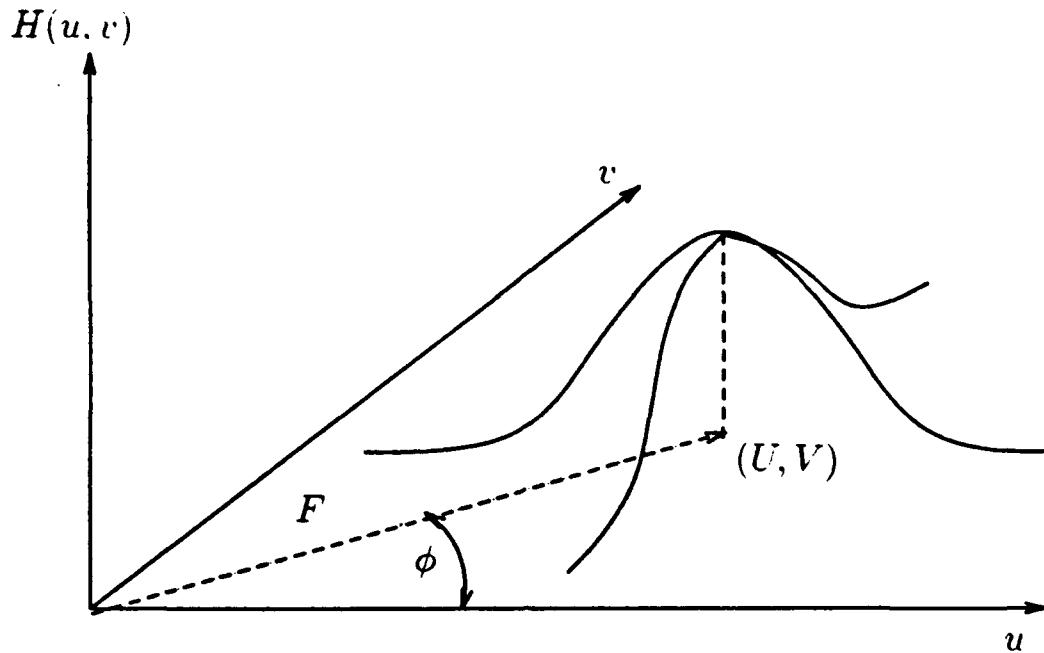


Fig. 3.2. Schematic of the frequency domain representation of a GEF.

λ of $g(x, y)$ differs from unity, the Gaussian is asymmetric with an orientation θ that generally differs from the orientation ϕ of the complex sinusoid. A schematic of the frequency domain representation of a GEF is shown in Fig. 3.2. When the Gaussian is circularly symmetric (i.e., $\sigma_x = \sigma_y = \sigma$), (U, V) in (3.2) can be expressed in polar coordinates. Then,

$$h(x, y) = g(x, y) e^{j\Omega x'} \quad (3.5)$$

where

$$\Omega = \sqrt{U^2 + V^2} \quad (3.6)$$

and $x' = x \cos \phi + y \sin \phi$. The corresponding Fourier transform of $h(x, y)$ is

$$H(u, v) = \exp\{-\sigma^2/2[(u' - \Omega)^2 + (v')^2]\}$$

where $(u', v') = (u \cos \phi + v \sin \phi, -u \sin \phi + v \cos \phi)$ are rotated frequency coordinates.

$H(u, v)$ is a Gaussian that is shifted radially Ω frequency units at an angle ϕ relative to the positive u axis. Thus, H acts as a bandpass filter with center frequency $(\Omega, 0)$ [relative to (u', v')] and a bandwidth controlled by σ .

The analysis in Chapter 5 will show that it is the bandpass nature of the GEF that is most essential for discriminating textural differences. Hence, since the aforementioned possibilities for filter functions - wavelet bases [62, 63], the difference of offset Gaussians [56, 64], and Gaussian derivatives [64] - also share this property, the choice of the GEF is not restrictive. Within the context of modeling human texture perception, Malik and Perona mentioned that the exact choice of a filter function was unimportant, and they chose the difference of offset Gaussians for computational simplicity and physiological plausibility [56]. Also, Bovik *et al.* have discussed the efficacy of bandpass filters for texture segmentation [53, 66].

The magnitude operation used in the Gabor filter (3.1) will now be discussed. Julesz has shown that purely linear mechanisms are inadequate to explain how humans perceive texture [67]. This point was further asserted by Malik and Perona [56] and can be illustrated with Fig. 3.3. Fig. 3.3a shows a uniformly textured image that is easily segmented by humans. If the homogeneous texture in Fig. 3.3b is added to Fig. 3.3a,

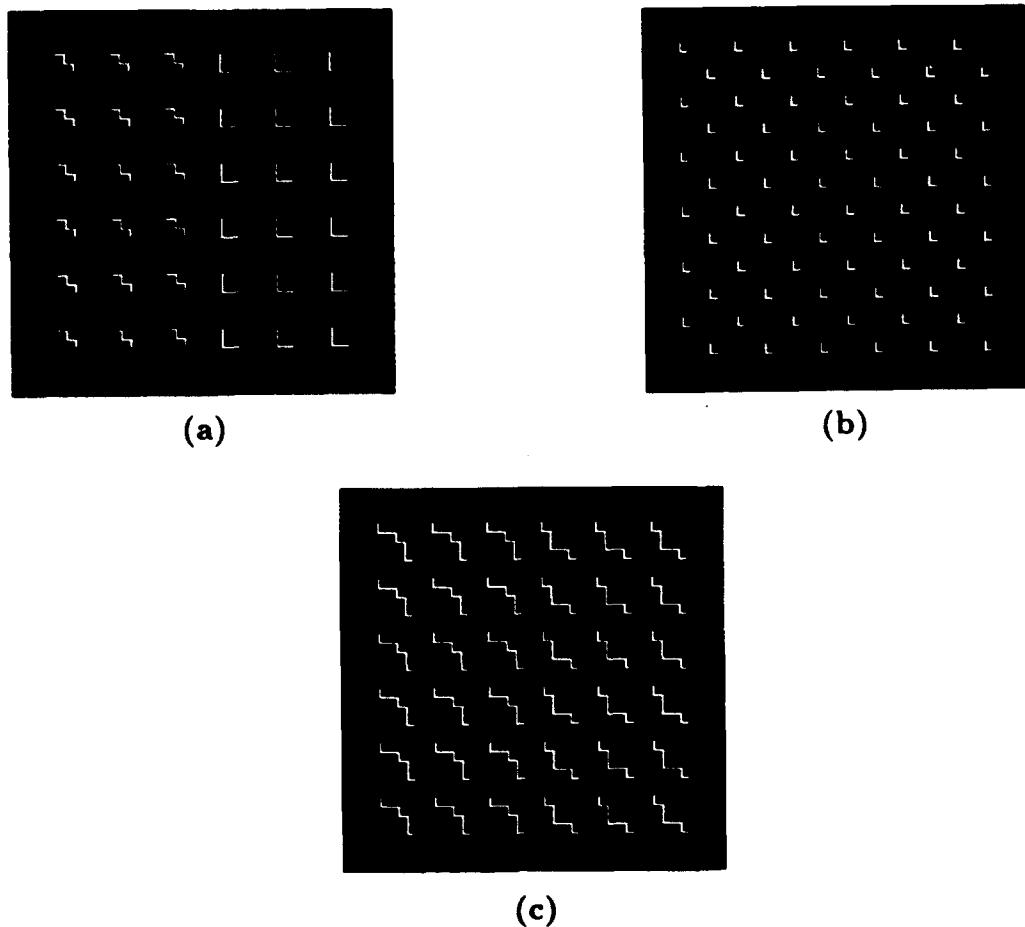


Fig. 3.3. Texture sequence demonstrating the need for a nonlinearity (from Malik and Perona [56]):
(a) Uniform texture pair-easily segmented.
(b) Homogeneous texture.
(c) Adding (a) to (b) produces a uniform texture pair that is difficult for a human to preattentively segment.

however, the resulting image (Fig. 3.3c) is difficult to segment. If purely linear mechanisms were involved in texture segmentation, one would expect Figs. 3.3a and 3.3c to be equally discriminable. Since they are obviously not, some form of nonlinearity must be present. Therefore, to simulate human texture perception, a nonlinearity is desirable. The magnitude operator introduces the desirable nonlinearity into the filter.

The convolution of an image with a GEF results in a complex-valued subimage. Bovik *et al.* have shown that the amplitude envelope of this subimage can be recovered by computing its magnitude and that the resulting amplitude envelope is useful for texture segmentation [53]. The magnitude operation has been frequently suggested in the literature [53, 55, 60, 68, 69, 70].

Note that the magnitude operation is not without flaw. Aside from being implausible neurophysiologically, Malik and Perona have shown that computing the magnitude makes it impossible to discriminate certain texture pairs [56]. Appendix A analytically verifies this assertion but then goes on to show that if mimicking human perception is not essential, then a wide range of textures can be segmented without using a nonlinearity. In spite of shortcomings, the magnitude computation provides a convenient analysis tool and serves as a benchmark for comparing alternatives.

Chapter 4

1-D Analysis

Using a mathematically defined 1-D texture model, this chapter analytically shows that applying properly configured GEF-based filters to textured images produces output discontinuities in the neighborhood of texture boundaries - this can be used to segment the image. Depending on the nature of the texture difference, this output discontinuity exhibits certain characteristic signatures. If two adjacent textures differ in local spatial-frequency content, this signature exhibits a *step change* at the location of the texture boundary. If the two adjacent textures differ only in a phase shift, the signature exhibits a *valley* at the location of the texture boundary. The following sections develop this theory in detail. First, the texture model is defined. Then, analysis shows under what conditions the various types of signatures occur.

Although the 1-D model has significant limitations, it leads to a simple analytical development providing useful insight. In Chapter 5, a more realistic 2-D model is presented, leading to more informative results at the expense of more complex analyses.

4.1 The 1-D Texture Model

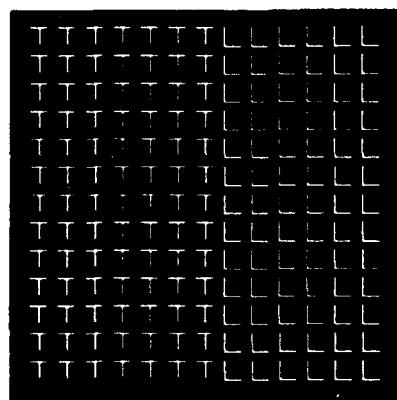
Although researchers have not agreed on a precise definition for texture, several descriptions have been proposed [12, 14, 71, 72]. Many textures can be described as a collection of similar but not necessarily identical primitive objects arranged in some repeating pattern. Based on this notion, texture will be modeled as a collection of simple

objects called *texels*. Groups of similar texels form regions of homogeneous texture. A textured image consists of two or more regions where texture differences between regions are induced by varying the type and/or organization of the texels. Using Rao's terminology, this approach can represent a variety of texture types, including strongly-ordered, weakly-ordered, and compositional textures [12]. This analysis does not address disordered textures, which due to their lack of structure, cannot be accurately modeled as a collection of texels. Experimental results in Chapter 8 indicate, however, that even disordered textures can be effectively discriminated/segmented with the filter-based approach.

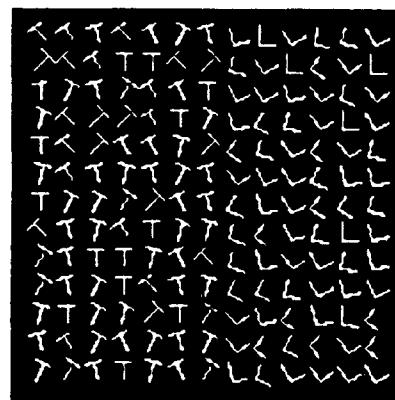
For convenience, textured images are divided into two levels of complexity: *uniform* and *nonuniform*. For uniform textures, all texels within a region are identical in shape and orientation and are spaced uniformly (e.g., Fig. 4.1a). For nonuniform textures, the texels within a region may vary randomly in orientation, and the position and shape of the texels may be perturbed (e.g., Fig. 4.1b).

To develop a simple mathematical model based on this description, an image i is constructed consisting of two textured regions 1 and 2 with the texels uniformly spaced. For simplicity, assume that the image is uniform in the y direction (i.e., $\forall y$, $i(x, y) = i(x)$). This reduces the analysis to one dimension. In 1-D, a simple texel can be modeled as

$$t(x) = f(x)\Pi_{\Delta x}(x)$$



(a)



(b)

Fig. 4.1. Examples of two levels of texture complexity: (a) uniform and (b) nonuniform.

where Π is a gate function with width Δx

$$\Pi_{\Delta x}(x) = \begin{cases} 1, & |x| < \frac{\Delta x}{2} \\ 0, & |x| \geq \frac{\Delta x}{2} \end{cases}$$

and f is some real-valued function of x . Consider two simple texels represented as amplitude-modulated gate functions

$$t_1(x) = \cos(\omega_1 x + \phi_1) \Pi_{\Delta x}(x) \quad (4.1)$$

$$t_2(x) = \cos(\omega_2 x + \phi_2) \Pi_{\Delta x}(x) \quad (4.2)$$

A 1-D textured region i_k , $k = 1, 2$, can be formed from a collection of N equally spaced texels t_k as follows:¹

$$i_k(x) = t_k(x) * \sum_{l=1}^N \delta(x - l\Delta h)$$

Each texel within i_k is a truncated sinusoid with frequency ω_k and phase angle ϕ_k . Thus, ω_k can be viewed as a local spatial frequency (i.e., local to a texel), which occurs at regular intervals throughout the region. i_k can therefore be characterized by this local frequency, and so ω_k is referred to as the *texture frequency* of textured region i_k .

A 1-D “image” i consisting of two nonoverlapping textured regions can be constructed as follows:

$$i(x) = i_1(x) + i_2(x - N\Delta h) \quad (4.3)$$

Assume for simplicity that the texel spacing is the same in both regions, and that each

¹To avoid technical difficulties in the analysis, i_k is allowed to take on positive or negative values.

region contains N texels. Note that when $\omega_1 \neq \omega_2$, i consists of two regions that differ in local spatial-frequency content. Texels t_1 and t_2 also contain phase components ϕ_1 and ϕ_2 . When $\phi_1 \neq \phi_2$ the texels will differ in phase. When i consists of two regions that have identical local spatial-frequency content but their texels differ in phase, a discontinuity in image phase occurs at the region boundary. This condition will be called a *texture-phase* difference. Subsequent analysis will show that a difference in local spatial-frequency content between textured regions causes a step change in Gabor-filter output, whereas a texture-phase difference produces a valley in the filter output.

4.2 Step Signature

The step signature is characterized by a step change in the Gabor-filter output m at the boundary between two textured regions. It occurs whenever there is a difference in average local spatial-frequency content between two regions. The occurrence of a step can be demonstrated by analyzing the Gabor filter response to an image consisting of regions with different texels.

Consider two simple texels with identical phase components (i.e., $\phi_1 = \phi_2 = 0$ in (4.1) and (4.2)).

$$t_1(x) = \cos(\omega_1 x) \Pi_{\Delta x}(x) \quad (4.4)$$

$$t_2(x) = \cos(\omega_2 x) \Pi_{\Delta x}(x) \quad (4.5)$$

The Fourier transforms of t_1 and t_2 are given by

$$T_1(\omega) = \frac{\Delta x}{2} \left[\text{sinc} \left(\frac{(\omega - \omega_1)\Delta x}{2} \right) + \text{sinc} \left(\frac{(\omega + \omega_1)\Delta x}{2} \right) \right]$$

$$T_2(\omega) = \frac{\Delta x}{2} \left[\text{sinc} \left(\frac{(\omega - \omega_2)\Delta x}{2} \right) + \text{sinc} \left(\frac{(\omega + \omega_2)\Delta x}{2} \right) \right]$$

where $\text{sinc}(x) \triangleq \sin(x)/x$. The Fourier transform of the image i in (4.3) is

$$I(\omega) = I_1(\omega) + I_2(\omega) \quad (4.6)$$

where

$$\begin{aligned} I_1(\omega) &= T_1(\omega) \sum_{l=1}^N e^{-j\omega l \Delta h} \\ I_2(\omega) &= T_2(\omega) \sum_{l=N+1}^{2N} e^{-j\omega l \Delta h} \end{aligned}$$

A Gabor filter (3.1) is now applied to i . Assume that the center frequency of a 1-D GEF equals ω_1 , the texture frequency for region i_1 . The Fourier transform for this 1-D GEF is

$$H_{\omega_1}(\omega) = e^{-\frac{\sigma^2}{2}(\omega - \omega_1)^2} \quad (4.7)$$

Thus, the GEF is a bandpass filter with center frequency ω_1 and bandwidth controlled by σ . Assume also that $1/\sigma$ is approximately equal to the main lobe width of T_1 and T_2 (i.e., $\frac{1}{\sigma} \approx \frac{4\pi}{\Delta x}$). This ensures that the bandwidth of the filter includes most of the energy of T_1 around ω_1 . Let us also assume that $\omega_1 \gg 0$ so that the sinc functions making up T_1 (centered at ω_1 and $-\omega_1$) do not significantly overlap. Finally, assume that ω_1 and

ω_2 are sufficiently separated (e.g., assume $|\omega_1 - \omega_2| > \frac{6}{\sigma}$). Then, applying $H_{\omega_1}(\omega)$ to the image (4.6) yields

$$\hat{I}(\omega) = H_{\omega_1}(\omega)I(\omega) \approx H_{\omega_1}(\omega) \frac{\Delta x}{2} \text{sinc}\left(\frac{(\omega - \omega_1)\Delta x}{2}\right) \sum_{l=1}^N e^{-j\omega l \Delta h}$$

Since H_{ω_1} is a function of $\omega - \omega_1$, we can define

$$F(\omega - \omega_1) \triangleq H_{\omega_1}(\omega) \frac{\Delta x}{2} \text{sinc}\left(\frac{(\omega - \omega_1)\Delta x}{2}\right) \quad (4.8)$$

Hence, the spatial-domain form of the filtered image is given by

$$i(x) = \mathcal{F}^{-1}[\hat{I}(\omega)] \approx \mathcal{F}^{-1}\left[F(\omega - \omega_1) \sum_{l=1}^N e^{-j\omega l \Delta h}\right] \quad (4.9)$$

Now,

$$F(\omega - \omega_1) e^{-j\omega x_0} \leftrightarrow f(x - x_0) e^{j\omega_1(x - x_0)} \quad (4.10)$$

is a Fourier transform pair. Therefore,

$$\hat{i}(x) = \sum_{l=1}^N f(x - l\Delta h) e^{j\omega_1(x - l\Delta h)} \quad (4.11)$$

where $f(x) = \mathcal{F}^{-1}[F(\omega)]$.

The complex exponential in (4.11) will cause oscillations in \hat{i} if it is not eliminated. Now suppose $\omega_1 \Delta h = n2\pi$ for some integer n (the implications of this assumption will be discussed shortly). Then, $\forall l$, the complex exponential reduces to $e^{j\omega_1 x}$ and $\hat{i}(x) = e^{j\omega_1 x} \sum_{l=1}^N f(x - l\Delta h)$.

Let us now examine f . The function f is awkward to deal with analytically, but we can get an intuitive feel for its shape. Referring to (4.8), we see that F is the product of a Gaussian and a sinc function. Since the inverse Fourier transform of a Gaussian is a Gaussian and the inverse Fourier transform of a sinc is a gate function, their multiplication in the frequency domain is equivalent to the convolution of a gate with a Gaussian in the spatial domain. For σ large relative to the texel spacing Δh , f is a greatly blurred gate function (i.e., a gate function with tapered shoulders). Then the sum of offset f 's approaches a constant, say C (actually a DC value with some ripple), over the range $0 \leq x \leq N\Delta h$. Thus,

$$\hat{i}(x) \approx \begin{cases} Ce^{j\omega_1 x} & \text{if } 0 \leq x \leq N\Delta h \\ 0 & \text{otherwise} \end{cases}$$

To complete application of the Gabor filter, we compute the magnitude of \hat{i} :

$$m(x) = G_f(\hat{i}(x)) = |\hat{i}(x)| \approx \begin{cases} C & \text{if } 0 \leq x \leq N\Delta h \\ 0 & \text{otherwise} \end{cases}$$

This implies that the output of the Gabor filter is approximately a constant value over region 1, where the texture frequency matches the filter center frequency, and zero over region 2. Thus, the Gabor filter output will be in the form of a step, with the transition occurring in the vicinity of the texture boundary.

We now return to the issue of assuming that $\omega_1\Delta h = n2\pi$ for some integer n . If we substitute $2\pi f_1$ for ω_1 , this equation reduces to $f_1 = n/\Delta h$. Since Δh is the texel spacing, this is equivalent to saying that the filter's center frequency is a multiple of

the frequency of occurrence of the texels. Thus, for a Gabor filter to produce a step signature with a small amount of ripple, f_1 should be a multiple of the reciprocal of the texel spacing.

The previous analyses show that f_1 is subject to two constraints: (1) f_1 should equal one of the texture frequencies, and (2) f_1 should be a multiple of the reciprocal of the texel spacing. For the cosine texture used in this discussion, satisfying both of these constraints is not always possible, since the texel spacing is not necessarily related to either texture frequency. For more complex textures, however, the local spatial-frequency content of the texels is typically broadband. In that case, the goal is to select a local 2-D frequency component (both radial frequency and orientation) that differs significantly in energy between the texels of different regions. This choice of frequencies makes it easier to satisfy both constraints.

One additional comment should be made regarding σ . We have assumed it to be large to reduce output ripple. However, it must not be made too large or we will lose accuracy in locating the region boundary. If σ is too large, then the tail of the last gate function will excessively extend into the adjacent region. This will cause error in estimating the region boundary. Thus, the choice of σ is a tradeoff between the amount of output ripple and texture-boundary resolution.

4.3 Valley Signature

The magnitude operation in the Gabor filter (3.1) discards the phase of the GEF-filtered image, resulting in a loss of information. (Appendix A discusses the issue of phase and alternatives to magnitude computation.) This section shows that certain

phase differences can be detected without using the phase component. This is in contrast to the approach of Bovik *et al.* [53], where phase information is extracted explicitly by demodulation of the channel phase component (cf. [53, 66, 68]). The approach is to design a suitable Gabor filter that detects discontinuities in the filter output m caused by abrupt changes in texture phase.

Consider the textured image of Fig. 1.3b. The two uniform regions are identical but offset vertically (the offset can also be horizontal). Thus the Fourier-transform magnitudes of the two regions are identical, but their respective phase characteristics differ. This type of texture difference will be referred to as a *texture-phase difference*. (This phenomena could equivalently be viewed as a collection of different texels near the texture boundary, but analysis suggests that a difference-in-phase interpretation is more appropriate.) Whereas a difference in local spatial-frequency content between textured regions causes a step change in the filter output, a texture-phase difference produces a valley in the output m .

To show how the valley signature can arise, again define two simple texels, t_1 and t_2 as in (4.1) and (4.2), and let $\omega_1 = \omega_2$, and $\phi_1 = 0$. Then, t_1 and t_2 are amplitude-modulated gate functions having the same frequency ω_1 but differing in phase:

$$t_1(x) = \cos(\omega_1 x) \Pi_{\Delta x}(x) \quad (4.12)$$

$$t_2(x) = \cos(\omega_1 x + \phi) \Pi_{\Delta x}(x) \quad (4.13)$$

where $\phi = \phi_2$. Thus, the Fourier transforms of t_1 and t_2 are given by

$$\begin{aligned} T_1(\omega) &= \frac{\Delta x}{2} \left[\text{sinc} \left(\frac{(\omega - \omega_1)\Delta x}{2} \right) + \text{sinc} \left(\frac{(\omega + \omega_1)\Delta x}{2} \right) \right] \\ T_2(\omega) &= \frac{\Delta x}{2} \left[\text{sinc} \left(\frac{(\omega - \omega_1)\Delta x}{2} \right) e^{-j\phi} + \text{sinc} \left(\frac{(\omega + \omega_1)\Delta x}{2} \right) e^{j\phi} \right] \end{aligned}$$

Let i again be a 1-D image consisting of two nonoverlapping equally spaced collections of t_1 and t_2 , as in (4.3). The Gabor-filter output for i will be derived next. The Fourier transform of i is

$$I(\omega) = T_1(\omega) \sum_{l=1}^N e^{-j\omega l \Delta h} + T_2(\omega) \sum_{l=N+1}^{2N} e^{-j\omega l \Delta h}$$

Again assume that the GEF is narrowband, centered at ω_1 , and that $\omega_1 \gg 0$. Then applying the filter H_{ω_1} , as defined in (4.7), to the image approximately retains only terms containing $\omega - \omega_1$. That is, after filtering

$$\hat{I}(\omega) = H_{\omega_1}(\omega) I(\omega) \approx F(\omega - \omega_1) \left[\sum_{l=1}^N e^{-j\omega l \Delta h} + e^{-j\phi} \sum_{l=N+1}^{2N} e^{-j\omega l \Delta h} \right] \quad (4.14)$$

where F is given by (4.8). From (4.10),

$$\hat{i}(x) \approx \sum_{l=1}^N f(x - l\Delta h) e^{j\omega_1(x - l\Delta h)} + e^{-j\phi} \sum_{l=N+1}^{2N} f(x - l\Delta h) e^{j\omega_1(x - l\Delta h)}$$

Again assume that $f_1 = \omega_1/(2\pi) = n/\Delta h$ for some integer n ; i.e., the GEF's center frequency is a multiple of the reciprocal of the texel spacing. Then, $\forall l$ the complex

exponentials within (4.14) reduce to $e^{j\omega_1 x}$. Thus,

$$\hat{i}(x) \approx e^{j\omega_1 x} \sum_{l=1}^N f(x - l\Delta h) + e^{j(\omega_1 x - \phi)} \sum_{l=N+1}^{2N} f(x - l\Delta h) \quad (4.15)$$

By the definition of the image i in (4.3), we know that the texture discontinuity occurs in the vicinity of $x = x_d = N\Delta h + \frac{\Delta h}{2}$. Let us compare the value of $\hat{i}(x_d)$ to values of $i(x)$ at points far removed from x_d . First, recall from before that $F(\omega - \omega_1)$ is a Gaussian multiplied by a sinc and resembles a gate function with tapered shoulders. Now, at x_d , the sum of f 's from the left sum becomes

$$f\left(\frac{\Delta h}{2}\right) + f\left(\frac{3\Delta h}{2}\right) + f\left(\frac{5\Delta h}{2}\right) + \dots \quad (4.16)$$

and the corresponding terms from the right sum are

$$f\left(\frac{-\Delta h}{2}\right) + f\left(\frac{-3\Delta h}{2}\right) + f\left(\frac{-5\Delta h}{2}\right) + \dots$$

Observe that f is symmetric about $\Delta h/2$ and that the dominant contribution to the sums are $f(\Delta h/2)$ and $f(-\Delta h/2)$, which are equal. Thus,

$$\hat{i}(x_d) \approx (f(\Delta h/2) + \epsilon) [e^{j\omega_1 x_d} + e^{j(\omega_1 x_d - \phi)}]$$

where ϵ represents the sum of the less significant terms. Computing the complex magnitude completes the application of the Gabor filter. Dropping terms involving ϵ gives

$$m(x_d) = G_f(i(x_d)) = |i(x_d)| \approx |f(\Delta h/2)| \cdot |S|$$

where

$$S = \sqrt{(\cos(\omega_1 x_d) + \cos(\omega_1 x_d - \phi))^2 + (\sin(\omega_1 x_d) + \sin(\omega_1 x_d - \phi))^2}$$

Note that $|S| < 2$, $\forall \phi$ such that $|\phi| > 0$. Hence,

$$m(x_d) = G_f(i(x_d)) < 2|f(\Delta h/2)|$$

Consider now a position $x \ll x_d$. Contributions to i will then be predominantly from the left sum in (4.15), which is expanded in (4.16). In this case, we have

$$i(x) \approx (2f(\Delta h/2) + \epsilon)e^{j\omega_1 x}$$

After computing the magnitude and dropping ϵ terms

$$m(x) = G_f(i(x)) = |i(x)| \approx |2f(\Delta h/2)| \quad (4.17)$$

Similarly, for $x \gg x_d$

$$i(x) \approx (2f(\Delta h/2) + \epsilon)e^{j(\omega_1 x - \phi)}$$

$m(x)$ is the same as in (4.17). Thus, we see that, for a filter tuned to the texture frequency, the value of the filter output m at the discontinuity is less than it is at other points. Changes in texture phase can thus be detected by locating valleys in the filter output m . In certain cases the discontinuity in filter output is a ridge rather than a valley. The 1-D texture model, however, is insufficient to derive the ridge signature. In the next chapter, a 2-D texture model is presented that allows for a more detailed analysis of the step and valley signatures and explains the origin of the ridge signature.

Chapter 5

2-D Analysis

The previous 1-D analysis showed that the application of a properly tuned Gabor filter to a textured image produces either a step or valley signature at texture boundaries. Using a 2-D texture model, similar to one proposed by Clark and Bovik [68], this chapter provides a more detailed development of the step and valley signatures and explains how the ridge signature can occur at a texture-phase discontinuity. The analysis also demonstrates the existence of certain signature anomalies called overshoot and undershoot and shows how they originate. The more general 2-D model also allows for evaluation of asymmetric filters (i.e., $\sigma_x \neq \sigma_y$ in (3.3)) and provides concrete guidelines for selecting Gabor-filter parameters. These issues are discussed in detail in Section 6.1. Although quantitative analysis is still limited to uniform textures, the 2-D model provides for a qualitative understanding of nonuniform textures. Section 5.2.3 discusses nonuniform textures and describes a fourth signature type resulting from the texel variation inherent in these textures.

5.1 2-D Texture Model

Section 4.1 developed a 1-D texture model based on collections of texels. This section presents a more robust 2-D version of that model. As in Section 4.1, an image i is formed from two uniform textures i_1 and i_2 . For the time being, assume that the two textures i_1 and i_2 consist of texels t_1 and t_2 that differ. Later, as necessary, these conditions will be varied.

Define a texel $t_1(x, y)$ as any real deterministic function that has a Fourier transform $T_1(u, v)$ that exists (singularity functions, such as impulses, are allowed to appear in $T_1(u, v)$). A uniform texture i_1 made up of an array of texels t_1 can be represented by

$$i_1(x, y) = t_1(x, y) * \sum_{k,l} \delta(x - k\Delta x, y - l\Delta y)$$

where Δx is the texel period in x , Δy is the texel period in y , and the Fourier transform of i_1 is

$$I_1(u, v) = \frac{4\pi^2}{\Delta x \Delta y} T_1(u, v) \sum_{k,l} \delta\left(u - \frac{2\pi k}{\Delta x}, v - \frac{2\pi l}{\Delta y}\right) \quad (5.1)$$

I_1 consists of a collection of weighted impulses whose signal energy are concentrated at the discrete set of frequencies $(2\pi k/\Delta x, 2\pi l/\Delta y)$. These frequencies will be referred to as the *harmonics* of I_1 . A uniform textured region with limited spatial extent \hat{i}_1 can be formed from i_1 by

$$\hat{i}_1(x, y) = \Pi_{r,s}(x, y) i_1(x, y) \quad (5.2)$$

where

$$\Pi_{r,s}(x, y) = \begin{cases} 1, & |x| < \frac{r}{2}, |y| < \frac{s}{2} \\ 0, & \text{otherwise} \end{cases}$$

is the 2-D gate function. Region \hat{i}_1 has support $r \times s$ and is centered at $(0, 0)$. The Fourier transform \hat{I}_1 of \hat{i}_1 is

$$\hat{I}_1(u, v) = \mathcal{F}[\hat{i}_1(x, y)] = \frac{1}{2\pi} [S(u, v) * I_1(u, v)] \quad (5.3)$$

where

$$S(u, v) = \mathcal{F}[\Pi_{r,s}(x, y)] = sr \operatorname{sinc}(ur/2) \operatorname{sinc}(vs/2) \quad (5.4)$$

and $\operatorname{sinc}(x) \triangleq \sin(x)/x$. Consider a second uniform texture i_2 made up of texels t_2 , where $t_2(x, y)$ is again any real deterministic function that has a Fourier transform $T_2(u, v)$ that exists (singularity functions again will be allowed in $T_2(u, v)$). Then,

$$i_2(x, y) = t_2(x, y) * \sum_{k,l} \delta(x - k\Delta x, y - l\Delta y) \quad (5.5)$$

A uniform textured region i_2 of support $r \times s$ and centered at $(r, 0)$ is given by

$$\hat{i}_2(x, y) = \Pi_{r,s}(x - r, y) i_2(x, y) \quad (5.6)$$

Then,

$$\hat{I}_2(u, v) = \mathcal{F}[\hat{i}_2(x, y)] = \frac{1}{2\pi} [S(u, v) e^{-jur}] * I_2(u, v) \quad (5.7)$$

where I_2 is similar to I_1 in (5.1), except that T_1 is replaced by T_2 . The regions i_1 and i_2 can be combined to form a finite-extent textured image

$$i(x, y) = i_1(x, y) + i_2(x, y) \quad (5.8)$$

Thus, i consists of two adjacent nonoverlapping textured regions i_1 and i_2 . See Fig. 5.1. The image i is spatially limited as a rectangular function to make analysis tractable (e.g., well-defined sinc functions, such as (5.4), occur frequently during the subsequent analysis). Also, a spatially limited i conforms to a real-world image setting. Clark and

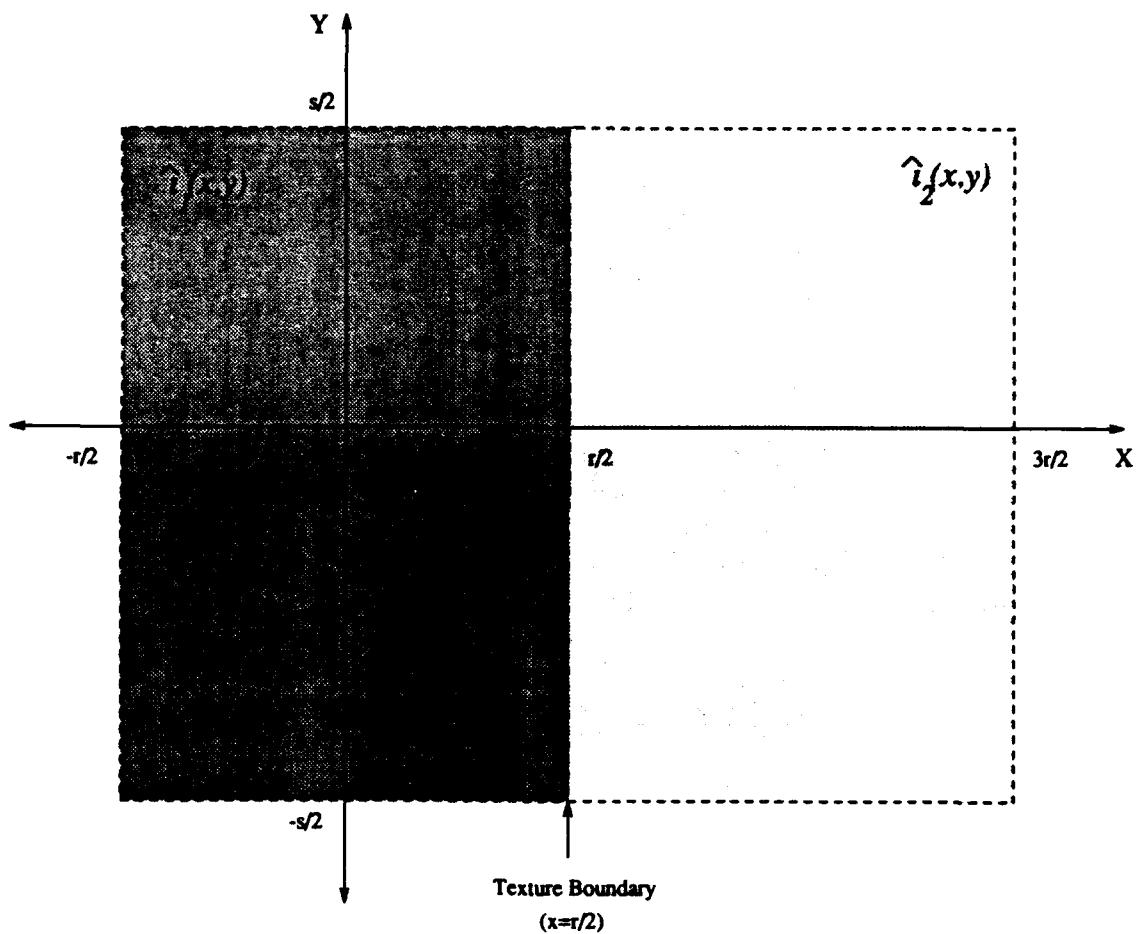


Fig. 5.1. Bipartite textured image model. Image $i(x,y)$ has support $2r \times s$ and is centered about $(r/2, 0)$. Texture $i_1(x,y)$ is made up of texels $t_1(x,y)$ and $i_2(x,y)$ is made up of texels $t_2(x,y)$.

Bovik employed a similar model, but their analysis used general indicator functions [68]. The analysis presented here leads to somewhat more tractable results and also more easily leads to an understanding of specific filter-output behavior.

Now,

$$\mathcal{F}[i(x, y)] = I(u, v) = \hat{I}_1(u, v) + \hat{I}_2(u, v) \quad (5.9)$$

where (evaluating (5.3) and (5.7)),

$$\hat{I}_1(u, v) = \frac{2\pi}{\Delta x \Delta y} \sum_{k,l} S_{k,l} T_1 \left(\frac{2\pi k}{\Delta x}, \frac{2\pi l}{\Delta y} \right) \quad (5.10)$$

$$\hat{I}_2(u, v) = \frac{2\pi}{\Delta x \Delta y} \sum_{k,l} S_{k,l} e^{-jr(u - \frac{2\pi k}{\Delta x})} T_2 \left(\frac{2\pi k}{\Delta x}, \frac{2\pi l}{\Delta y} \right) \quad (5.11)$$

$$S_{k,l} = S(u - 2\pi k/\Delta x, v - 2\pi l/\Delta y) \quad (5.12)$$

and $S(u, v)$ is given by (5.4). Or

$$I(u, v) = \frac{2\pi}{\Delta x \Delta y} \sum_{k,l} S_{k,l} \left(T_1 \left[\frac{2\pi k}{\Delta x}, \frac{2\pi l}{\Delta y} \right] + e^{-jr(u - \frac{2\pi k}{\Delta x})} T_2 \left[\frac{2\pi k}{\Delta x}, \frac{2\pi l}{\Delta y} \right] \right) \quad (5.13)$$

Observe that \hat{I}_1 consists of a collection of scaled 2-D sinc functions centered at the harmonics $(2\pi k/\Delta x, 2\pi l/\Delta y)$. The amplitude of the sinc $S_{k,l}$ at harmonic $(2\pi k/\Delta x, 2\pi l/\Delta y)$ is proportional to the value of the Fourier transform of the texel T_1 evaluated at that harmonic. \hat{I}_2 also consists of a collection of scaled 2-D sinc functions centered at the harmonics. The amplitudes of the sincs for \hat{I}_2 , however, are proportional to T_2 rather than T_1 , and their phase components are influenced by a complex phase factor. Thus, by (5.13), I is a sum of scaled sincs $S_{k,l}$. Each sinc consists of a component from each

texture region. Or, more colloquially, each (k, l) component of I consists of a *pair of sincs*, one for each texture.

Thus, the *texture segmentation problem* is to find the boundary separating regions i_1 and i_2 in image i . Per the model's construction, the boundary separating these two textures is the line segment given by $x = r/2$ and $|y| < s/2$. The goal is to understand how the Gabor filter (3.1) will help in locating this boundary.

5.2 Characterizing Gabor-Filter Outputs

This section shows analytically that the application of Gabor filters to textured images produces outputs that exhibit discontinuities in the neighborhood of texture boundaries. This is shown within the context of the texture model defined in the Section 5.1. The analysis begins with those texture configurations that produce a step signature and is followed by an analysis of texture types that produce a valley or ridge signature.

5.2.1 Textures Consisting of Different Texels: Step Signature

This section derives conditions when the application of Gabor filter (3.1) to a uniformly textured image produces a step signature. The step signature is characterized by a step change in the Gabor-filter output m at the boundary between two textured regions. This signature type occurs when a properly tuned Gabor filter is applied to a uniformly textured image that contains two textures whose constituent texels t_1 and t_2 differ.

To derive this result, consider the outcome of applying a Gabor filter (3.1) to the

textured-image model I in (5.9) (or equivalently i in (5.8)). The goal is to design a filter that enables “easy” localization of the texture boundary. Analytically, the approach is to design a Gabor filter that passes the image energy centered about one harmonic (\hat{k}, \hat{l}) . This is equivalent to passing one and only one scaled sinc $S_{\hat{k}, \hat{l}}$ in (5.13), where the sinc draws contributions from each texture; i.e., design a filter that passes one sinc pair occurring at some harmonic (\hat{k}, \hat{l}) . Each sinc in the pair represents a gate function in the spatial domain. Each gate coincides with one of the two region boundaries, and the difference in gate amplitude is proportional to the amplitude difference between the two sincs (i.e., $|T_1 - T_2|$). By filtering out a sinc pair whose sincs differ significantly in amplitude, a filter output is produced that is approximately constant within a region, but differs between regions, thus forming a step signature.

Designing a Gabor filter involves specifying the five parameters $(U, V, \sigma_x, \sigma_y, \theta)$ of the GEF H in (3.4). To pass the single sinc-pair at harmonic indices (\hat{k}, \hat{l}) , the center frequency (U, V) of H is specified as $U = 2\pi\hat{k}/\Delta x$, $V = 2\pi\hat{l}/\Delta y$. The bandwidth of H , determined by (σ_x, σ_y) , is then selected so that H passes most of the image energy centered about harmonic (\hat{k}, \hat{l}) , while also largely rejecting the image energy at adjacent harmonics. Since harmonic spacing is proportional to texel spacing $(\Delta x, \Delta y)$, the ratios $(\sigma_x/\Delta x, \sigma_y/\Delta y)$ determine this filter characteristic. Clearly, the choice of $(\sigma_x/\Delta x, \sigma_y/\Delta y)$ is a tradeoff between attenuation of the desired harmonic and a rejection of adjacent harmonics. The consequences of this tradeoff are discussed in Section 6.1.

Applying H to I gives

$$I_f(u, v) = H(u, v)I(u, v)$$

Since H has been designed to pass only those frequency components in the neighborhood of (U, V) , we can write

$$I_f(u, v) \approx \frac{2\pi}{\Delta x \Delta y} H(u, v) S(u - U, v - V) \{T_1 + T_2 e^{-jr(u-U)}\} \quad (5.14)$$

where T_1 and T_2 are abbreviations for $T_1(U, V)$ and $T_2(U, V)$. Observing that H in (3.4) is a function of $u - U$ and $v - V$, the function S_f is defined as

$$S_f(u - U, v - V) \triangleq H(u, v) S(u - U, v - V) \quad (5.15)$$

where

$$\mathcal{F}^{-1}[S_f(u - U, v - V)] = s_f(x, y) e^{j(Ux+Vy)}$$

and

$$s_f(x, y) = \mathcal{F}^{-1}[S_f(u, v)] \quad (5.16)$$

By substituting S_f into (5.14), the inverse Fourier transform of I_f can be expressed as

$$i_f(x, y) = \frac{2\pi}{\Delta x \Delta y} e^{j(Ux+Vy)} [T_1 s_f(x, y) + T_2 s_f(x - r, y)] \quad (5.17)$$

Computing the magnitude of i_f completes the application of the Gabor filter and gives

$$m(x, y) = |i_f(x, y)| = \frac{2\pi}{\Delta x \Delta y} \sqrt{A + B + C} \quad (5.18)$$

where

$$\begin{aligned} A &= |T_1|^2 s_f^2(x, y) \\ B &= |T_2|^2 s_f^2(x - r, y) \\ C &= (T_1^* T_2 + T_1 T_2^*) s_f(x, y) s_f(x - r, y) \end{aligned}$$

(It can be seen from (3.4), (5.4), (5.15), and (5.16) that s_f is real.)

To understand the behavior of m , we first need to determine s_f . S_f equals a sinc multiplied by a Gaussian. Thus, in the spatial domain, s_f can be expressed as the convolution of a Gaussian with a gate function:

$$s_f(x, y) = \int_{-s/2}^{s/2} \int_{-\tau/2}^{\tau/2} g(x - \alpha, y - \beta) d\alpha d\beta \quad (5.19)$$

$$s_f(x - r, y) = \int_{-s/2}^{s/2} \int_{\tau/2}^{3\tau/2} g(x - \alpha, y - \beta) d\alpha d\beta \quad (5.20)$$

where g is the Gaussian (3.3). The quantity m can now be evaluated by examining its behavior at the texture boundary and at points far removed from the boundary (or equivalently points within the interiors of each texture). Assume that the region width τ in the x direction is large relative to σ_x , and the region height s in the y direction is large relative to σ_y . Then, for points away from the textured image's outer boundary and left of the texture boundary (i.e., $|y| \ll s/2$ and $|x| \ll \tau/2$ for points in region 1), $s_f(x, y) \approx 1$ and $s_f(x - r, y) \approx 0$ per (5.19) and (5.20). Then $m \approx \frac{2\pi}{\Delta x \Delta y} |T_1|$. Similarly, for points to the right of the texture boundary in region 2, $m \approx \frac{2\pi}{\Delta x \Delta y} |T_2|$.

Now, at the texture boundary ($x = r/2$), the filter output m is

$$\begin{aligned} m(r/2, y) &\approx \frac{2\pi}{\Delta x \Delta y} \sqrt{|T_1|^2/4 + |T_2|^2/4 + (T_1^* T_2 + T_1 T_2^*)/4} \\ &\approx \frac{\pi}{\Delta x \Delta y} \sqrt{(T_1 + T_2)(T_1 + T_2)^*} = \frac{\pi}{\Delta x \Delta y} |T_1 + T_2| \end{aligned} \quad (5.21)$$

since $s_f(x, y)$ and $s_f(x - r, y)$ both $\approx 1/2$ at $x = r/2$. Now suppose that T_1 and T_2 are both real and positive. Then $m(r/2, y)$ becomes the average of values far to the right and left of the texture boundary, and (5.18) can be rewritten as

$$\begin{aligned} m(x, y) &= \frac{2\pi}{\Delta x \Delta y} (T_1^2 s_f^2(x, y) + T_2^2 s_f^2(x - r, y) + 2T_1 T_2 s_f(x, y) s_f(x - r, y))^{1/2} \\ &= \frac{2\pi}{\Delta x \Delta y} (T_1 s_f(x, y) + T_2 s_f(x - r, y)) \end{aligned}$$

Observing that $s_f(x - r, y) \approx 1 - s_f(x, y)$, we see that m is a linear function of s_f . Since s_f is the integral of a Gaussian, its shape is similar to a sigmoid function. Thus m is also shaped like a sigmoid in the neighborhood of the texture boundary. Assuming that $|T_1| \neq |T_2|$, m is given by a constant value

$$A_1 = \frac{2\pi}{\Delta x \Delta y} |T_1| \quad (5.22)$$

over region 1 and a constant value

$$A_2 = \frac{2\pi}{\Delta x \Delta y} |T_2| \quad (5.23)$$

over region 2 with a sigmoid transition between regions; i.e.,

$$m(x, \cdot) \approx \begin{cases} A_1, & x < r/2 \\ \text{sigmoidal transition} \\ \text{from } A_1 \text{ to } A_2, & x \text{ near } r/2 \text{ (texture boundary)} \\ A_2, & x > r/2 \end{cases}$$

Thus m resembles a step function with the transition occurring near the texture boundary.

Suppose now that T_1 and T_2 are negative or complex. Then m can take on values $< \min(A_1, A_2)$ or $> \max(A_1, A_2)$ near the texture boundary. These possibilities are referred to as *undershoot* and *overshoot* respectively. To see how undershoot can occur, (5.21) shows that near the texture boundary m is proportional to $(T_1 + T_2)$. Thus, if T_1 and T_2 are negative or complex, the magnitude of their sum can be less than the magnitude of either component. Overshoot can occur if the Gabor-filter center frequency (U, V) is not equal to one of the harmonics of I . The phenomena of undershoot and overshoot need not overly complicate the detection of the texture boundary. They are illustrated in Chapter 8 and discussed analytically in Appendix B.

5.2.2 Textures Consisting of Identical Texels, but Exhibiting a Texture-Phase Difference: Valley and Ridge Signatures

This section shows that the application of a Gabor filter to a textured image exhibiting a texture-phase difference (as defined in Section 4.3) produces a *valley* in the Gabor-filter output m when the GEF is properly tuned: if an improperly tuned GEF is used, a *ridge* occurs in m at the texture boundary.

5.2.2.1 Valley Signature

Again, the goal is to design a filter that enables easy localization of the texture boundary. Analytically, the procedure is to design a Gabor filter that passes the image energy centered about one harmonic $(2\pi\hat{k}/\Delta x, 2\pi\hat{l}/\Delta y)$. This is equivalent to passing one and only one sinc pair centered about some harmonic $(2\pi\hat{k}/\Delta x, 2\pi\hat{l}/\Delta y)$. In this case, the amplitudes of the sincs are identical. The offset regions, however, produce a phase shift ψ (given in (5.26)) between the sincs, resulting in a drop in filter output given by (5.29) near the texture boundary.

First, the texture model of Section 5.1 is modified to fit the texture-phase-difference scenario. Define a texel t_1 as before and construct a uniform textured region i_1 as in (5.2). Define a second texel t_2 equal to t_1 but shifted δx in the x direction and δy in the y direction, where $0 < \delta x < \Delta x$ and $0 < \delta y < \Delta y$. Then

$$t_2(x, y) = t_1(x - \delta x, y - \delta y)$$

A uniform texture i_2 whose texels are periodic in x and y can be constructed from this texel as shown in (5.5), and a uniform textured region i_2 of support $r \times s$ and centered at $(r, 0)$ can be formed from i_2 as in (5.6). Thus a uniform textured image i that exhibits a texture-phase difference at $x = r/2$ can be formed similar to (5.8):

$$i(x, y) = i_1(x, y) + i_2(x, y)$$

$\mathcal{F}[i(x, y)]$ is then similar to (5.9):

$$I(u, v) = \hat{I}_1(u, v) + \hat{I}_2(u, v)$$

$\hat{I}_1(u, v)$ is given by (5.10), but $\hat{I}_2(u, v)$ differs from (5.11), since

$$T_2(u, v) = T_1(u, v)e^{-j(u\delta x + v\delta y)}$$

Thus

$$I(u, v) = \frac{2\pi}{\Delta x \Delta y} \sum_{k,l} S_{k,l} T_1 \left(\frac{2\pi k}{\Delta x}, \frac{2\pi l}{\Delta y} \right) \left\{ 1 + e^{-jr(u-2\pi k/\Delta x)} e^{-j2\pi(k\delta x/\Delta x + l\delta y/\Delta y)} \right\} \quad (5.24)$$

Let the GEF H have center frequency (U, V) , where $U = 2\pi\hat{k}/\Delta x$ and $V = 2\pi\hat{l}/\Delta y$ for some (\hat{k}, \hat{l}) , and select $(\sigma_x/\Delta x, \sigma_y/\Delta y)$ as in Section 5.2.1. Applying H to I approximately passes only the sinc-pair centered at (U, V) :

$$\begin{aligned} I_f(u, v) &= H(u, v)I(u, v) \\ &\approx \frac{2\pi}{\Delta x \Delta y} T_1(U, V) H(u, v) S(u - U, v - V) \left\{ 1 + e^{-jr(u-U)} e^{-j2\pi(\hat{k}\delta x/\Delta x + \hat{l}\delta y/\Delta y)} \right\} \end{aligned}$$

Defining S_f as in (5.15), the inverse Fourier transform of I_f is

$$i_f(x, y) = \frac{2\pi}{\Delta x \Delta y} T_1(U, V) e^{-j(Ux+Vy)} \left[s_f(x, y) + s_f(x - r, y) e^{-j2\pi(\hat{k}\delta x/\Delta x + \hat{l}\delta y/\Delta y)} \right] \quad (5.25)$$

Let

$$\psi = 2\pi(\hat{k}\delta x/\Delta x + \hat{l}\delta y/\Delta y) \quad (5.26)$$

ψ represents the *total relative phase shift* between regions 1 and 2. Computing the magnitude of i_f completes the application of the Gabor filter and gives

$$\begin{aligned} m(x, y) &= C \left| s_f(x, y) + s_f(x - r, y) e^{-j\psi} \right| \\ &= C \sqrt{s_f^2(x, y) + s_f^2(x - r, y) + 2s_f(x, y)s_f(x - r, y) \cos \psi} \end{aligned} \quad (5.27)$$

where

$$C = \frac{2\pi}{\Delta x \Delta y} |T_1(U, V)| \quad (5.28)$$

Consider the behavior of m . Assume that a phase shift occurs; i.e., $\forall(\hat{k}, \hat{l}), \psi \neq$ multiple of 2π or, equivalently, choose some (\hat{k}, \hat{l}) such that $\cos \psi \neq 1$. (This holds, because of the restrictions placed earlier on δx and δy .) The image does not exhibit a phase discontinuity in the y direction; so in subsequent analyses, it will be assumed that y is far removed from the image's outer boundaries (i.e., $|y| \ll s/2$). Consider m over three regions. The first region consists of those values of x such that $|x| \ll r/2$ (i.e., points in region 1 far from both the texture boundary and the image's outer boundary - see Fig. 5.1). In this case, (5.19) and (5.20) indicate that, for r large relative to σ_x , $s_f(x, y) \approx 1$, and $s_f(x - r, y) \approx 0$. Thus, from (5.27), $m \approx C$. The second region consists of those values of x for $r/2 \ll x \ll 3r/2$ (points in region 2). In this case, $s_f(x - r, y) \approx 1$, $s_f(x, y) \approx 0$, and $m \approx C$. The last region is in the neighborhood of the texture boundary

($x \approx r/2$). At $x = r/2$, for large r , $s_f(x, y) \approx s_f(x - r, y) \approx 1/2$, and from (5.27),

$$m(r/2, y) \approx C \sqrt{0.5(1 + \cos \psi)} \quad (5.29)$$

Summarizing,

$$m(x, \cdot) \approx \begin{cases} C, & x < r/2 \\ C \sqrt{0.5(1 + \cos \psi)}, & x \text{ near } r/2 \text{ (texture boundary)} \\ C, & x > r/2 \end{cases}$$

Note that near the texture boundary ($x \approx r/2$), $m(x, \cdot) \leq C$ is required, because of the weighting factor $\sqrt{0.5(1 + \cos \psi)}$. Thus, for this situation, a *valley signature occurs near the texture boundary*.

When no phase shift exists between the two regions, the transition takes on its maximum value C , which is the same value as for points far removed from the transition. This is expected, since without a texture-phase shift, the two regions are indistinguishable. If $\psi = \pi$ (maximum texture-phase difference between two regions), the value of m at the texture boundary is 0 and a *minimum valley results*. Note from (5.26) that the depth of the valley depends on the ratios $\delta x / \Delta x$ and $\delta y / \Delta y$. These ratios represent the amount of texture-phase shift in x and y relative to the texel periods. Thus, the greater the texture-phase shift, the deeper the valley.

5.2.2.2 Ridge Signature

With the Gabor filter designed as indicated above, a ridge signature cannot occur near the texture boundary. It can be shown, however, that *if the Gabor filter is tuned to a frequency other than an harmonic, a ridge is produced.*

To see this, consider the frequency domain representation of the GEF-filtered image

$$I_f(u, v) = H(u, v)I(u, v)$$

In this case, let $U = 2\pi\hat{k}/\Delta x + \delta U$, $V = 2\pi\hat{l}/\Delta y + \delta V$, where δU and δV are chosen so that $2\pi\hat{k}/\Delta x < U + \delta U < 2\pi(\hat{k} + 1)/\Delta x$ and $2\pi\hat{l}/\Delta y < V + \delta V < 2\pi(\hat{l} + 1)/\Delta y$. Now, the inverse Fourier transform i_f has the same form as in (5.25), but now s_f represents a gate function convolved with a GEF having center frequency $(\delta U, \delta V)$ rather than a gate function convolved with a simple Gaussian. Convolving a gate function with a GEF produces a complex quantity. So, s_f becomes complex. Thus, computing the magnitude of i_f as in (5.27) produces

$$m(x, y) = C \sqrt{P_r P_r^* + P_0 P_0^* + P_r P_0^* e^{-j\psi} + P_0 P_r^* e^{j\psi}} \quad (5.30)$$

where $P_t = s_f(x - t, y)$. It can be shown that if $\delta U \neq 0$ or $\delta V \neq 0$, then the complex terms P_0 and P_r constructively interfere with each other to produce an increase in filter output near the texture boundary, thus forming a *ridge signature*. The height of the ridge depends on the texture phase shift ψ . The details leading to this result follow.

Since s_f is now the result of convolving a gate function with a GEF, we have (cf.

(5.19), (5.20))

$$s_f(x, y) = \int_{-s/2}^{s/2} \int_{-r/2}^{r/2} h(x - \alpha, y - \beta) d\alpha d\beta \quad (5.31)$$

$$s_f(x - r, y) = \int_{-s/2}^{s/2} \int_{r/2}^{3r/2} h(x - \alpha, y - \beta) d\alpha d\beta \quad (5.32)$$

h is the GEF in (3.2) and can be represented by the sum of its real and imaginary parts:

$$h(x, y) = h_r(x, y) + h_i(x, y) \quad (5.33)$$

where

$$h_r(x, y) = g(x', y') \cos[2\pi(\delta U x + \delta V y)] \quad (5.34)$$

$$h_i(x, y) = -jg(x', y') \sin[2\pi(\delta U x + \delta V y)] \quad (5.35)$$

Again consider m over three image regions. The first region is defined by $|x| \ll r/2$ (points well inside region 1). From (5.31) and (5.33),

$$s_f(x, y) = \int_{-s/2}^{s/2} \int_{-r/2}^{r/2} [h_r(x - \alpha, y - \beta) + h_i(x - \alpha, y - \beta)] d\alpha d\beta \quad (5.36)$$

For large r , the integral of h_r approaches some constant γ_1 , and the integral of h_i (since it is an odd function) approaches zero. Also for large r , $s_f(x - r, y)$ approaches zero. Thus $s_f(x, y) \approx \gamma_1$, $s_f(x - r, y) \approx 0$, and by (5.30) $m(x, y) \approx \gamma_1 C$. A similar argument holds for points in region 2 ($r/2 \ll x$).

The third region is the transition near $x = r/2$. At $x = r/2$, (5.36) can be

rewritten as

$$s_f(r/2, y) = \int_{-s/2}^{s/2} \int_0^r [h_r(\alpha, y - \beta) + h_i(\alpha, y - \beta)] d\alpha d\beta$$

Again assume r to be large. In this case, the imaginary component does not go to zero.

Since the integral of the real component equals γ_1 , we can write

$$\gamma_1 = \int_{-s/2}^{s/2} \int_0^r h_r(\alpha, y - \beta) d\alpha d\beta \quad (5.37)$$

For the imaginary component, let

$$\gamma_2 = \int_{-s/2}^{s/2} \int_0^r h_i(\alpha, y - \beta) d\alpha d\beta \quad (5.38)$$

Similarly for $s_f(x - r)$,

$$s_f(-r/2, y) = \int_{-s/2}^{s/2} \int_{-r}^0 [h_r(\alpha, y - \beta) + h_i(\alpha, y - \beta)] d\alpha d\beta$$

Therefore, at $x = r/2$,

$$s_f(x, y) = \gamma_1 + j\gamma_2$$

$$s_f(x - r, y) = \gamma_1 - j\gamma_2$$

and (5.30) reduces to

$$m(r/2, y) = C \sqrt{2\gamma_1^2 + 2\gamma_2^2 + (\gamma_1^2 - \gamma_2^2 + j2\gamma_1\gamma_2)e^{j\psi} + (\gamma_1^2 - \gamma_2^2 - j2\gamma_1\gamma_2)e^{-j\psi}}$$

$$= C \sqrt{2\gamma_1^2 + 2\gamma_2^2 + 2(\gamma_1^2 - \gamma_2^2) \cos \psi - 4\gamma_1\gamma_2 \sin \psi} \quad (5.39)$$

Note that when $\psi = 0$ or a multiple of 2π (i.e., no texture phase shift), $m = \gamma_1 C$ as it does at points far removed from the transition. For a ridge to occur, m at the transition $x = r/2$ must be greater than $\gamma_1 C$:

$$\begin{aligned} \gamma_1 C &< m(r/2, y) \\ &< C \sqrt{2\gamma_1^2 + 2\alpha^2\gamma_1^2 + 2(\gamma_1^2 - \alpha^2\gamma_1^2) \cos \psi - 4\alpha\gamma_1^2 \sin \psi} \\ &< \gamma_1 C z(\alpha, \psi) \end{aligned} \quad (5.40)$$

where we let γ_2 be represented as

$$\gamma_2 = \alpha\gamma_1 \quad (5.41)$$

for some constant α , and

$$z(\alpha, \psi) = \sqrt{2 + 2\alpha^2 + 2(1 - \alpha^2) \cos \psi - 4\alpha \sin \psi} \quad (5.42)$$

where it can be shown that $z \geq 0$. Thus, summarizing:

$$m(x, \cdot) \approx \begin{cases} \gamma_1 C, & x < r/2 \\ \gamma_1 C z(\alpha, \psi), & x \text{ near } r/2 \text{ (texture boundary)} \\ \gamma_1 C, & x > r/2 \end{cases}$$

and a ridge occurs when $z > 1$. Note that (5.42) is not restricted to determining the ridge height. For $z < 1$, a valley occurs, and (5.42) can be used to approximate its depth.

Table 5.1. z as a function of α with ψ_{\max} (in degrees) chosen to maximize z .

α	ψ_{\max}	z
0.1	-11	1.005
0.2	-23	1.020
0.3	-33	1.044
0.4	-44	1.077
1.5	-113	1.803
2.8	-141	2.973

Setting $\partial z / \partial \psi = 0$ gives the value of ψ that maximizes z :

$$\psi_{\max} = \tan^{-1} \left[\frac{-2\alpha}{1 - \alpha^2} \right]$$

Table 5.1 gives z versus α for ψ_{\max} (in degrees).

Observe that as $\alpha \rightarrow 0$, $z \rightarrow 1$ from above, and $\psi_{\max} \rightarrow 0$. Thus, as long as $\gamma_2 > 0$, a ridge can occur if ψ is sufficiently close to ψ_{\max} . From (5.38), it is clear that $\gamma_2 > 0$ whenever the center frequency $(\delta U, \delta V)$ of h_i is nonzero and finite; i.e., *when the Gabor-filter center frequency is not an harmonic of i* .

As we have just seen, if an improperly tuned Gabor filter is applied to an image exhibiting a texture-phase difference, a ridge signature can occur at the texture boundary. While generating a ridge is not the ideal result, it can still be useful for texture segmentation. To be able to perform texture segmentation, though, the ridge must be reasonably strong. The ridge's strength depends upon its height, which depends on ψ , the texture-phase shift, and on γ_2 in (5.38). Appendix C gives a method for computing ridge height.

5.2.3 Nonuniform Textures

Unlike uniform textures, nonuniform textures consist of texels that can undergo perturbations in position, orientation, and shape. These perturbations make analysis more difficult. Although Clark and Bovik have investigated the effect of texel-position variability [68], they did not consider perturbations in texel orientation and shape. Analyzing the effects of texel orientation and shape perturbations is much more difficult, especially in the general case, because the results depend strongly on individual texel characteristics. Thus, this thesis does not provide a quantitative analysis for nonuniform textures, but instead presents qualitative arguments and experimental results (Chapter 8). Chapter 7 gives more discussion on nonuniformly textured images.

The experimental results indicate that the signatures obtained for uniformly textured images also can occur for nonuniformly textured images. In contrast to uniform textures, though, a nonuniformly textured image cannot be represented in the frequency domain simply as a 2-D impulse train. The perturbations possible in the texels introduces more frequency components beyond just simple harmonics. The net result is that the Gabor-filter output signatures obtained for nonuniformly textured images typically exhibit many local output variations. (See, for example, Fig. 5.2.) In spite of these output variations, the results of Chapter 8 demonstrate that Gabor-filter outputs derived from nonuniformly textured images can be useful for texture segmentation.

In addition to the three signature types derived for uniform textures, nonuniformly textured images can exhibit a fourth signature type. Consider, for example, two regions that have the same average spatial-frequency content, but the spatial-frequency variation

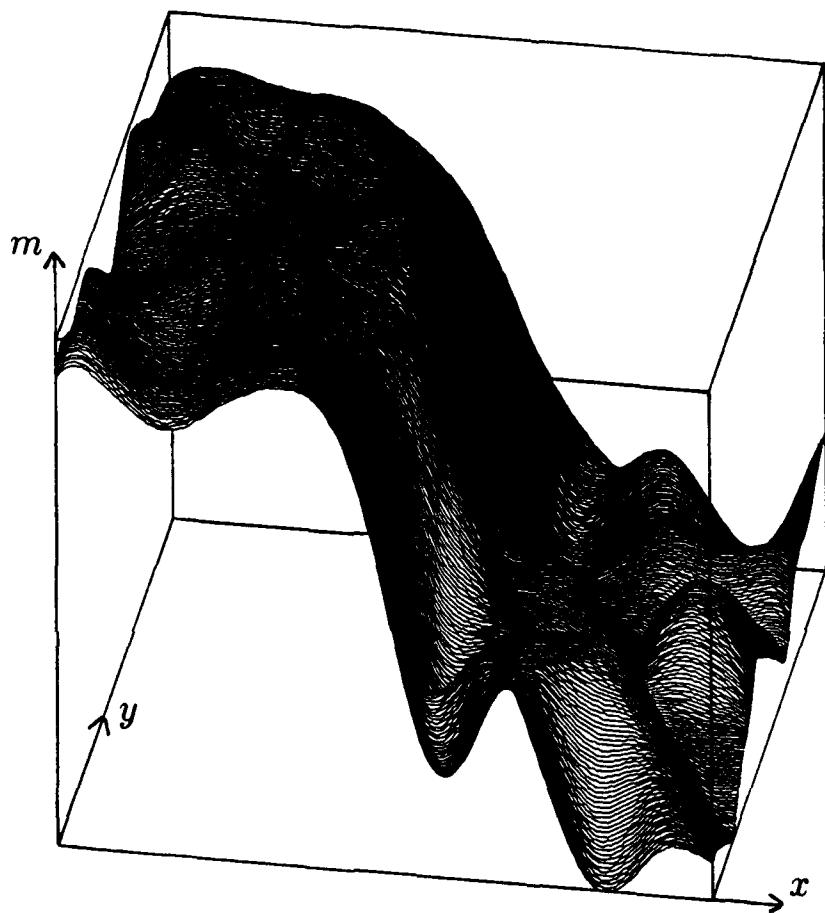


Fig. 5.2. Example Gabor-filter output from a nonuniformly textured image.

is different between regions. In this case, the local variation in the Gabor-filter output m will differ in the two regions. This fourth form of discontinuity will be referred to as a *step change in the average local output variation* of m . Fig. 1.4 gives an example of this type of signature.

With a step, valley, or ridge discontinuity, standard image-segmentation techniques could be used to locate the discontinuity. This is not the case, however, with a change in average local output variation without some further filtering. One possible solution is to transform this quantity into a change in mean value. Turner [60] encountered this problem and suggested using a bandpass filter for detecting such local changes. For our situation, this would involve applying a second Gabor filter to the output m of the first. If the variations within two regions have similar frequency content, then the second filter output would be proportional to the magnitude of the variation. Thus, a difference in average output variation would translate into a difference in mean for the second filter's output.

Another simple method for transforming a difference in average local output variation into a difference in mean is to perform the following operation on m :

$$\text{LPF}\{|m(x, y) - \mu_m|\} \quad (5.43)$$

Here μ_m is the mean value of m (DC component) and LPF is a low pass filter. This method does not make any assumptions on the frequency content of the input. Other methods are possible.

Chapter 7 examines nonuniformly textured images in depth and gives a numerical

technique for measuring the discriminability of arbitrary pairs of textures. This measure of discriminability, then, leads to a method for selecting appropriate Gabor-filter parameters for discriminating any given texture pair.

Chapter 6

Gabor-Filter Parameter Guidelines

Section 5.2 showed that when a *properly tuned* Gabor filter is applied to a textured image, distinct output signatures arise at the texture boundaries. This chapter describes how to select filter parameters for a properly tuned Gabor filter. Section 6.1 provides parameter guidelines based on previous analyses. Then Section 6.2 discusses parameter constraints that arise when a bank of filters is to be designed. For nonuniform and natural textures, however, the guidelines provided by analyses are only approximately correct; so in Chapter 7, an algorithm is presented for numerically determining filter parameters for an arbitrary texture pair.

6.1 Guidelines for Selecting Filter Parameters

Based on the analyses of Chapters 4 and 5, this section provides guidelines for selecting the parameters for a properly tuned Gabor filter. Chapter 8 provides design examples and image-segmentation results. This section assumes that the given image contains two uniformly textured regions, whose constituent texels are t_1 and t_2 (as defined in Section 5.1) and whose texel spacings are $(\Delta x_1, \Delta y_1)$ and $(\Delta x_2, \Delta y_2)$. For convenience, Table 6.1 summarizes the parameter selection criteria.

Table 6.1. Gabor-filter design criteria for processing a textured image containing two textured regions, $R_1(t_1, \Delta x_1, \Delta y_1)$ and $R_2(t_2, \Delta x_2, \Delta y_2)$.

<p>A) Uniform Textures, constituent texels t_1 and t_2 differ and/or texel spacings $(\Delta x_1, \Delta y_1)$ and $(\Delta x_2, \Delta y_2)$ differ — Gabor-filter output: <i>Step Signature</i>.</p>
<p>$(\sigma_x, \sigma_y) \triangleq$ Spatial extent of GEF's Gaussian envelope.</p> <ul style="list-style-type: none"> a) Recommendation: $\sigma_x = \Delta x$, $\sigma_y = \Delta y$, where $\Delta x = \max(\Delta x_1, \Delta x_2)$; $\Delta y = \max(\Delta y_1, \Delta y_2)$. b) If $\Delta x_1 \neq \Delta x_2$ or $\Delta y_1 \neq \Delta y_2$, then GEF cannot be tuned to both harmonics and undershoot/overshoot can occur in Step Signature. c) (σ_x, σ_y) large relative to $(\Delta x, \Delta y)$ — output signatures <i>cleaner</i>. d) (σ_x, σ_y) small relative to $(\Delta x, \Delta y)$ — texture boundary better localized.
<p>$\lambda = \sigma_y / \sigma_x \triangleq$ aspect ratio of Gaussian envelope.</p>
<p>$(U, V) \triangleq$ GEF center frequency.</p> <ul style="list-style-type: none"> a) Recommendation: $U = 2\pi\hat{k}/\Delta x$, $V = 2\pi\hat{l}/\Delta y$, where $\hat{k}, \hat{l} = \arg \left\{ \max_{\hat{k}, \hat{l}} T_1(U, V) - T_2(U, V) \right\}$, \hat{k}, \hat{l} integers. b) Depends on texel spacing $(\Delta x, \Delta y)$ and differences between harmonics $(\frac{2\pi\hat{k}}{\Delta x}, \frac{2\pi\hat{l}}{\Delta y})$ of t_1 and t_2. c) Tune center frequency to harmonic exhibiting maximum difference. d) If not tuned precisely to an harmonic, undershoot/overshoot could result in output signature.
<p>$\theta \triangleq$ Orientation of Gaussian Envelope.</p> <ul style="list-style-type: none"> a) Recommendation: Select orientation along direction of texel-spacing lattice. b) Independent of $(\sigma_x, \sigma_y, U, V)$.
<p>B) Uniform Textures, Texture-Phase Difference (texels identical in two regions ($t_1 = t_2$)), texel spacing same in two regions $\Delta x_1 = \Delta x_2$, $\Delta y_1 = \Delta y_2$, and regions shifted relative to each other — Gabor-filter output: <i>Valley or Ridge Signature</i>.</p>
<p>(σ_x, σ_y) Design criteria as in Part A.</p>
<p>$\lambda = \sigma_y / \sigma_x$</p>
<p>(U, V)</p> <ul style="list-style-type: none"> a) Recommendation: Select (U, V) equal to an harmonic $(2\pi\hat{k}/\Delta x, 2\pi\hat{l}/\Delta y)$ as in Part A — produces <i>valley</i> signature. b) If $(U, V) \neq$ an harmonic — nonideal, but still discriminating, <i>ridge</i> signature results.
<p>θ Design criteria as in Part A.</p>
<p>C) Nonuniform Textures: Texels (t_1, t_2) differ and texels perturbed in position, orientation, and shape — Gabor-filter output: <i>Step, Valley, or Ridge Signature</i> with output variation or <i>Difference in Local Output Variation Signature</i>.</p>
<p>All filter parameters - Use guidelines in part A. - Use techniques of Chapter 7 for parameter selection.</p>

6.1.1 Texels in Two Textured Regions Differ

Assume that the texels t_1 and t_2 differ. Thus, the discussion focuses on the design of a Gabor filter tuned for producing a step-signature output. Also, assume for the time being that the texel spacings for the two textures are identical; i.e., $\Delta x_1 = \Delta x_2 = \Delta x$, $\Delta y_1 = \Delta y_2 = \Delta y$. This condition will later be relaxed.

The parameters to select for the Gabor filter are (σ_x, σ_y) , λ , (U, V) , and θ . Fig. 6.1 illustrates the relationship between filter size and texel spacing. The ellipse represents the one-standard-deviation contour of the Gaussian envelope of a Gabor filter; i.e., $\{x, y\}$, such that $(x/\sigma_x)^2 + (y/\sigma_y)^2 = 1$. The positioning of the ellipse at point (x, y) represents the position of the GEF when the Gabor-filter output m is computed at point (x, y) .

The choice of σ_x and σ_y is a tradeoff between Gabor-filter output variation and accurate boundary localization. When $\sigma_x > \Delta x$ and $\sigma_y > \Delta y$, the filter envelope encompasses multiple texels, regardless of its position in the image. Although the positions of the texels vary within the envelope as the filter progresses across the image, the Gabor-filter output m remains approximately constant over a region. If $\sigma_x \ll \Delta x$ or $\sigma_y \ll \Delta y$, the filter output depends on whether or not a texel occurs within the GEF envelope. This results in periodic Gabor-filter output variations throughout a region. To avoid significant output variation, $\sigma_x/\Delta x$ and $\sigma_y/\Delta y$ should both be large; i.e., the GEF's spatial extent should cover a number of texels. If (σ_x, σ_y) are large, though, near the texture boundary, the filter envelope will extend into both regions. This region overlap is what produces the sigmoid output transition described in Section 5.2.1. As (σ_x, σ_y) become larger, the transition becomes more gradual, making it more difficult to locate the

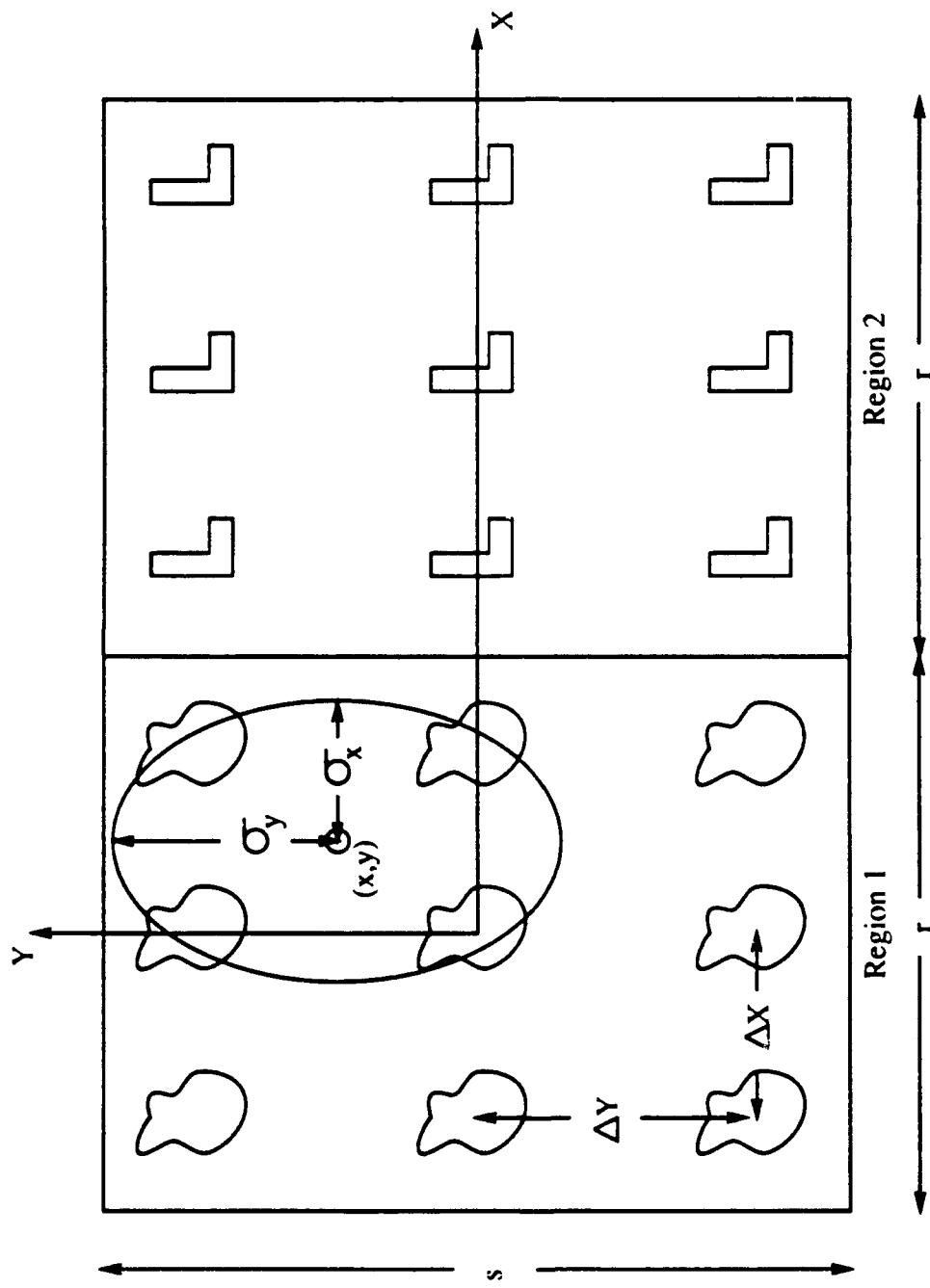


Fig. 6.1. Schematic representation of the application of a Gabor filter to a uniformly textured image. Image consists of two adjacent regions, each containing nine texels. The ellipse represents the application of a GEF at one point in the convolution (3.1).

texture boundary. Experimental evidence suggests that filter performance is relatively insensitive to these ratios. A good compromise is to set them to unity; i.e., $\sigma_x = \Delta x$, $\sigma_y = \Delta y$.

If the texel spacings in the x and y directions differ ($\Delta x \neq \Delta y$) [but the two textures still use the *same* spacing!], then using the aforementioned design criteria, $\sigma_x \neq \sigma_y$. Thus the filter's aspect ratio $\lambda = \sigma_y/\sigma_x \neq 1$, resulting in an *asymmetric* filter. For asymmetric filters, the orientation θ of the Gaussian in the GEF (3.2) becomes an issue. Based on the discussion above, the Gaussian should be oriented to encompass on the average as many texels as possible. If the texels are spaced over a rectangular lattice, the Gaussian should be oriented along the x and y axes (i.e., $\theta = 0$ or $\pi/2$). If the texels are not spaced over such a lattice but are situated relative to some rotated coordinate system (x', y') , then the Gaussian should be oriented along the rotated axes. The orientation of the complex sinusoid ϕ is determined by the Gabor-filter center frequency (U, V) , and thus by the analysis in Section 5.2, depends on the spectral differences between texels. Therefore, the choice of θ is in general independent of ϕ .

The choice of center frequency (U, V) depends on the texel spacing (which determines the harmonics) and on the spectral differences between texels at the harmonics. As discussed in Section 5.2.1, (U, V) should be set to the harmonic that differs most in power between the texels in the two regions. Although two texels might differ more at some nonharmonic frequency, using this frequency as a filter center frequency in general produces an output signature that exhibits overshoot and/or undershoot - such signatures have lower values within the textures than the values produced by a properly tuned filter. This is shown in Appendix B (compare (A_1, A_2) to (\hat{A}_1, \hat{A}_2)).

6.1.2 Texel Spacings Differ between Textured Regions

When texel spacing is the same in both regions, each texture has spectral energy centered about the same harmonics (cf. (5.13)), and a Gabor filter can be designed to produce step-signature outputs. If the texel spacings of the two regions differ, the harmonics from the different textures do not coincide. Since a Gabor-filter can be tuned to only one harmonic, signature distortion will result. In analyzing this distortion, note that the Gabor-filter operation (prior to computing the magnitude) is linear, allowing the study of each region independently. Assume that the Gabor-filter center frequency (U, V) equals an harmonic of region 1. Then, the analysis proceeds as for the step signature. The frequency coordinates for the nearest corresponding harmonic of region 2 can be written as $(U + \delta U, V + \delta V)$, where $(\delta U, \delta V)$ is the frequency offset between the harmonics of the two regions. Thus, the analysis of region 2 becomes analogous to that for the ridge signature. Combining the results for the two regions and computing the magnitude results in

$$m(x, y) = |i_f(x, y)| = 2\pi\sqrt{P_r P_r^* + P_0 P_0^* + P_r P_0^* + P_0 P_r^*} \quad (6.1)$$

where $P_0 = \frac{T_1(U, V)}{\Delta x_1 \Delta y_1} s_f(x, y)$, $P_r = \frac{T_2(U + \delta U, V + \delta V)}{\Delta x_2 \Delta y_2} s_f(x - r, y)$, and $s_f(x, y)$ and $s_f(x - r, y)$ are given by (5.19) and (5.32). For points far removed from the texture boundary, an analysis similar to that in Section 5.2.1 reveals that (6.1) produces a step signature. The analysis near the texture boundary, however, suffers from the same complications encountered in evaluating overshoot and undershoot (Appendix B). Although a detailed

analysis is impractical, the presence of the GEF integral in (6.1) suggests that overshoot and/or undershoot can be expected. An example in Chapter 8 corroborates this observation.

6.2 Parameter Constraints for Filter Banks

Section 6.1 provided guidelines for selecting filter parameters for individual filters. These guidelines, however, are based upon specific image characteristics. In general, though, these characteristics are not known *a priori* (assuming that ultimately we are striving for a truly autonomous texture-segmentation system), and thus it is difficult to choose appropriate Gabor-filter parameters. Instead, a collection of such filters must be specified (i.e., a *filter bank*), where each filter is tuned to a different frequency band, and collectively they span the range of frequencies expected in the input. These filters are then applied to the image (conceptually in parallel [56]), and their outputs are combined in a meaningful way, so as to partition the image into regions of homogeneous texture. Defining a filter bank involves specifying the number of filters within the filter bank and the parameters for each of these filters. The need to combine filter outputs imposes certain restrictions on these filters. These restrictions are discussed below.

6.2.1 A Constraint on σ

Assume for the moment that all textured images to be encountered have the same texel spacing and that only symmetric Gabor filters are to be used (i.e., $\sigma_x = \sigma_y = \sigma$). This section shows that if Gabor-filter outputs are to be compared, the corresponding Gabor filters must have equal values of σ .

Let us assume that the input image is periodic with period T in both directions. (This simplifies the frequency analysis without affecting the grayscale distribution in the regions of interest.) The image, then, can be represented by its complex Fourier series

$$i(x, y) = \sum_{n=-\infty}^{\infty} \sum_{m=-\infty}^{\infty} c_{m,n} e^{j\omega(nx+my)}$$

where $\omega = 2\pi/T$, and $c_{m,n}$ are the Fourier-series coefficients. Consider the GEF h of (3.5). For simplicity, assume that h is oriented along the x axis. Then $x' = x$. Convolution of i with h after separating the integrals yields

$$\hat{i}(x, y) = i(x, y) * h(x, y) = \frac{1}{2\pi\sigma^2} \sum_n \sum_m c_{m,n} J_1 \int_{-\infty}^{\infty} e^{-\frac{\alpha^2}{2\sigma^2}} e^{j\Omega\alpha} e^{j\omega n(x-\alpha)} d\alpha$$

where $J_1 = \int_{-\infty}^{\infty} e^{-\frac{\beta^2}{2\sigma^2}} e^{j\omega m(y-\beta)} d\beta$. The integral J_1 equals

$$\sigma\sqrt{2\pi} e^{j\omega my} e^{-\frac{(\omega m\sigma)^2}{2}}$$

and after rearranging terms

$$\hat{i}(x, y) = \frac{1}{\sqrt{2\pi}\sigma} \sum_n \sum_m c_{m,n} e^{j\omega(my+nx)} e^{-\frac{(\omega m\sigma)^2}{2}} \int_{-\infty}^{\infty} e^{-\frac{\alpha^2}{2\sigma^2}} e^{j\alpha(\Omega-\omega n)} d\alpha$$

The remaining integral reduces to $\sqrt{2\pi}\sigma e^{-\frac{\sigma^2(\Omega-\omega n)^2}{2}}$ and the final result is

$$\hat{i}(x, y) = \sum_m \sum_n c_{m,n} e^{j\omega(my+nx)} e^{-\frac{\sigma^2}{2}[(\omega m)^2 + (\Omega - \omega n)^2]} \quad (6.2)$$

Equation (6.2) shows that the resulting subimage \hat{i} is composed of the original

input with each term of the input being reduced by the factor $e^{-\frac{\sigma^2}{2}[(\omega_m)^2 + (\Omega - \omega_n)^2]}$. Two features of this equation should be emphasized. First, the only term in the input that is left unattenuated is the complex sinusoid oriented in the x direction (i.e., $m = 0$), with a frequency equal to the center frequency of the filter (i.e., $n\omega = \Omega$). Second, the parameter σ , combined with the frequency of the harmonic, controls the degree of attenuation of the harmonic. This suggests that it is not feasible to make comparisons between filter outputs, if σ is not the same for both filters.

For example, consider two filters with the same center frequency, but different σ 's. If these filters are applied to the same input, the filter with the smaller σ will usually produce a larger output. Unless the frequency distribution of the input is known, however, the amount of difference cannot be determined. That is, filter outputs cannot be normalized without knowing the frequency distribution of the input. Thus all filters within a given filter bank must have the same value of σ .

The choice of σ depends on the texel spacing; so images with different texel spacings (e.g., images at different scales), require filters with different values of σ . Since the output of filters with different σ 's cannot be reliably compared, multiple filter banks must be used, with each bank consisting of a collection of filters with the same σ . The idea is to partition the range of texel spacings into k intervals, specify the σ_k 's, and define a filter bank for each σ_k .

6.2.2 Other Parameter Constraints

Because the filters within a filter bank should span the expected 2-D frequency range of the input images, large texel spacings can present a problem. As the texel

spacing becomes larger, σ becomes larger, and thus, the bandwidth of the filters becomes narrower. This means that more filters are required to cover the same frequency band. Although there is an upper bound on the number of frequencies in the band (dictated by the number of image pixels sampled), the number could become very large. This problem can be circumvented by recalling that as an image increases in scale, its frequency content is compressed; i.e., $g(ax) \leftrightarrow \frac{1}{|a|}G(\omega/a)$. Images tend to have most of their energy around DC, with energy diminishing rapidly at the higher frequencies. This means that there exists a cutoff frequency f_c above which their energy is insignificant. Since increasing the size of an image results in a proportional compression of frequency, the net effect is a similar reduction in cutoff frequency. Even though the frequency spacing between filters decreases with increasing σ , the cutoff frequency decreases proportionally. Thus, the number of filters remains constant. Therefore, a fixed number of frequencies can be assigned to each bank of filters, without incurring a significant loss in energy.

Once σ has been specified (thus defining a filter bank for a particular interval of texel spacings), the frequency parameters (Ω, ϕ) must be specified for each filter within the bank. Since the frequency content of an image is not typically known *a priori*, the filters must cover the entire range of expected frequencies; however, because the image is sampled, the entire range of frequency harmonics (in both the u and v directions) is known. Given a pair of frequencies U and V , it is easy to compute Ω and ϕ : $\Omega = \sqrt{U^2 + V^2}$ and $\phi \triangleq \tan^{-1}(V/U)$. With this information, all possible values of Ω and ϕ can be determined. If the number of samples is large, however, it would be impractical to have a filter at every possible frequency. It might be sufficient to have significant overlap between adjacent filters to cover the desired frequency range and 360 degrees

of orientation. The amount of frequency and orientation overlap is determined by the center frequency spacing and the bandwidth of the filter, which is controlled by σ . One possible choice for filter spacing is to make the difference between center frequencies of adjacent filters equal to the half-peak bandwidth of one of the filters. A similar choice can be made for orientation overlap, and thus the number of required orientations can be determined. Bovik *et al.* [53] derived a half-peak orientation bandwidth for the GEFs. The radian bandwidth Z is defined as $Z = 2 \tan^{-1}[2\alpha/(\Omega\sigma)]$ where $\alpha = \sqrt{(\ln 2)/2}$. Thus, for each center frequency Ω , we define a set of filters, each with a different orientation parameter ϕ , and each ϕ spaced Z radians apart.

One popular filter configuration that is consistent with these constraints is the “rosette” pattern [63, 70, 73]. In the 2-D frequency plane, the rosette consists of overlapping filters whose center frequencies lie on concentric circles centered at the origin. This configuration spans 360 degrees of orientation and spans frequencies from DC upward to any desired resolution. One formulation that directly leads to this pattern are Gabor wavelets [62, 63]. An example of such a pattern is shown in Fig. 6.2. One limitation of the rosette pattern is that it does not allow independent selection of (σ_x, σ_y) and (U, V) ; however, in practice this is not a problem. Recall that (σ_x, σ_y) are related to the texel spacing. Since texel spacing changes with image scale and frequency content is proportional to scale, (σ_x, σ_y) and (U, V) are related. Thus independent selection of center frequency and filter size might not be necessary. Section 6.1.1 showed that asymmetric filters can be beneficial when the texel spacings differ in x and y . The rosette pattern, however, does not allow for varying filter asymmetry. Often in practice, though, smooth signatures can still be attained at some cost in boundary localization if the texel-spacing

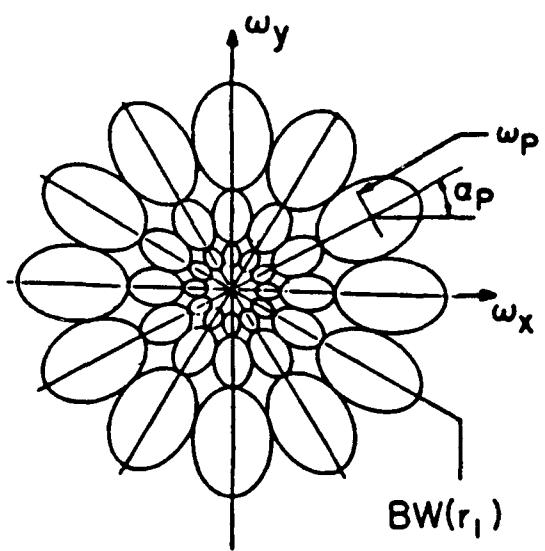


Fig. 6.2. Example of a "rosette" pattern of bandpass filters (from Porat and Zeevi [73]).

difference is disregarded. Thus, it appears that the rosette pattern is a plausible filter-bank configuration. Determining the number of filter banks, the number of filters in each filter bank, and the filter spacing in the 2-D frequency plane are topics for future work.

Chapter 7

Determining Filter Parameters for Nonuniform Textures

The parameter guidelines developed in Section 6.1, are only approximately correct for nonuniform and natural textures. Since σ_x , σ_y , and θ , depend primarily on texel organization, the guidelines for these parameters are still applicable. The frequency parameters (U, V) , however, are no longer simply related to the difference in the texel Fourier transforms. Thus, the methods of Section 6.1 are inadequate for determining (U, V) . Previous efforts in determining Gabor-filter frequency parameters have involved: (1) computing the Fourier transforms of the textures of interest and determining the most discriminating frequency [53], (2) using heuristics gleaned from studies of the human visual system [56, 60, 74], (3) performing a spectral decomposition on prototype texture elements for each texture of interest and noting where large differences occur [15, 55], and (4) *ad hoc* selection [62, 69]. As Section 7.3 later points out, these methods all have limitations.

This section develops an algorithm for determining the Gabor-filter frequency parameters for any given texture pair. For convenience, the algorithm will be referred to as GFFS (for Gabor-Filter Frequency Selection). Given instances of a texture pair of interest, GFFS searches the space of Gabor-filter center frequencies to determine the Gabor filter that provides the “greatest” discrimination between the two textures. Note that this is a supervised approach to frequency selection. The method provides an analytical tool for evaluating the segmentability of texture pairs using a single filter. The

remainder of the section elaborates on GFFS and compares it to previously proposed techniques for determining Gabor filter center frequencies. Chapter 8 shows experimental results demonstrating the efficacy of the new technique.

7.1 Overview of the GFFS Algorithm

As mentioned previously, the application of a Gabor filter to a textured image $i(x, y)$ can produce an output image $m(x, y)$ exhibiting some type of discontinuity at the texture boundaries (called signatures). This output then can be used to segment the image. *The problem is to find Gabor-filter parameters that will produce one of these discontinuities at the texture boundary.*

Depending on the texture pair and the filter parameters, different signature types can occur. The most common of these is the step signature (i.e., a step change in filter output $m(x, y)$). For the next several sections, it will be assumed that we wish to design a Gabor filter that produces the “best” step signature at a texture boundary. Section 7.4 elaborates on how the method can be extended to other signature types. The method for determining Gabor-filter parameters for producing a step signature will now be described.

The problem statement is the following. *Given a textured image consisting of known textured regions A and B, find the Gabor filter giving the largest step change at the texture boundary.* This Gabor filter is determined by the parameters σ_x , σ_y , U , V , and θ per (3.4), and must be found from among the space of all possible Gabor filters.

Chapter 6 provided guidelines for selecting σ_x , σ_y , and θ . Experience indicates that these guidelines are also effective for nonuniform and natural textures. This allows

us to use heuristic methods for determining these parameters (more on this in Section 7.2.3). Thus, the method reduces to determining the Gabor filter center frequencies (U, V). This is accomplished by essentially performing an exhaustive search over all possible frequencies.

In principle, the quality of a step signature is determined by the amplitude of the step. Previous analyses (Section 5.2) and experimental results (Chapter 8), however, show that the Gabor-filter output $m(x, y)$ resembles an ideal step only in special cases. More often, due to the inherent random structure within texture, the step is accompanied by considerable local variation. Thus, directly measuring the step amplitude is infeasible. Instead, stochastic decision theory is used to develop an alternative measure of step-signature quality.

Developing a measure of step-signature quality begins by modeling the Gabor-filter outputs from textured regions A and B as independent random variables, having pdf's p_A and p_B (Section 7.2.1). Then, for each σ_x , σ_y , and θ considered, the “best” Gabor-filter center frequencies (U, V) are determined as follows:

1. Apply a windowed Fourier transform (WFT) to a random set of points within each textured region A and B - this effectively gives information on the application of a family of Gabor filters to each of the random points (Section 7.2.2). Each filter in the family has a different center frequency (U, V). Collectively these center frequencies effectively span the frequency domain of the textured region. Using the computed WFT information, estimate parameters for the pdf's p_A and p_B (Section 7.2.1).

2. Using the estimated pdf's for textures A and B , apply a likelihood-ratio test to compute the probability of correctly determining from which region (either A or B) a Gabor-filter output value arose. This probability indicates the statistical difference between the Gabor-filter outputs in the two regions, and is used as a measure of step-signature quality.
3. The center frequency (U, V) producing the highest quality step signature is determined the "best" and used to design h in (3.2).

The complete algorithm is summarized below. For each σ_x , σ_y , and θ of interest, do the following:

1. For each textured region A and B ,
 - a. Form a randomly selected set s of points within the region.
 - b. For each point $(X, Y) \in s$, compute

$$F_{X,Y}(U, V) = \left| \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} i(x, y) g(x - X, y - Y) \exp[-j(Ux + Vy)] dx dy \right| \quad (7.1)$$

where g is the Gaussian (3.3) and i is the image. F is the windowed Fourier transform of i centered at (X, Y) and g is the window function. The computation of F , which is implemented as an $N \times N$ DFT, effectively applies a family of Gabor filters to the point (X, Y) , where the center frequencies (U, V) of the filters correspond to the $N \times N$ set of 2-D frequencies given by the DFT.

c. For each (U, V) , compute

$$\bar{\mu}(U, V) = \frac{\sum_s F_{X,Y}(U, V)}{\text{card}(s)} \quad (7.2)$$

$$\bar{\sigma}^2(U, V) = \frac{\sum_s (F_{X,Y}(U, V) - \bar{\mu}(U, V))^2}{\text{card}(s)} \quad (7.3)$$

where $\bar{\mu}(U, V)$ and $\bar{\sigma}^2(U, V)$ are the sample mean and sample variance for the values of F averaged over all points (X, Y) considered in step b above.

2. For each (U, V) , compute $P_E(U, V)$, the total probability of incorrectly classifying textures A and B , per (7.10). $P_E(U, V)$ gives a measure of step-signature quality for a textured image (containing textured regions A and B) filtered by a Gabor filter having parameters $(\sigma_x, \sigma_y, U, V, \theta)$.
3. The values of (U, V) corresponding to the minimum value of $P_E(U, V)$ is the “best” Gabor-filter center frequency.

After applying the procedure above, one “best” center frequency (U, V) is obtained for each set $(\sigma_x, \sigma_y, \theta)$ considered. Two options are now available: either pick the “best” center frequency for large values of (σ_x, σ_y) (as described in Section 7.2.3) and apply the Gabor filter in (3.1), or pick the “best” center frequency for small values of (σ_x, σ_y) and apply the modified Gabor filter ((3.1) followed by (7.15)) as discussed in Section 7.2.4.

Section 7.2 provides justification and comments on the various aspects of the GFFS algorithm. In particular, it (1) derives a measure of step-signature quality based on stochastic signal detection theory, (2) shows how the simultaneous application of a family of Gabor filters can be implemented efficiently using the windowed Fourier

transform, and (3) discusses the selection of σ_x and σ_y .

7.2 Algorithm Implementation Issues

7.2.1 Measuring Step-Signature Quality

Before Gabor-filter parameters can be evaluated, some measure of step-signature quality needs to be established. Since the location of the step transition presumably corresponds to the texture boundary, basing signature quality on accurate step-edge detection and localization is attractive. In Canny's development of an ideal step-edge detector, he shows that both the detection and localization of the step improves directly as A/n_0 increases, where A is the step amplitude and n_0 is the average noise amplitude [75]. For the GFFS algorithm, A is the mean difference in Gabor-filter output between regions, and n_0 corresponds to the local fluctuations in filter output within a region. If A/n_0 is defined as the signal-to-noise ratio (S/N) of the Gabor-filter output m , then the S/N seems to be a reasonable basis for signature quality.

A measure of step-signature quality based on the S/N can be derived by viewing step detection as a stochastic signal detection problem, where the goal is to minimize the error in erroneously classifying one signal (textured region) as another. Within this framework, the Gabor-filter output within a given textured region is considered to be a random variable. Although the distribution of this random variable is, in general, unknown, it will be assumed that its distribution can be approximated by a Gaussian over the range of probabilities of interest. (Clearly it is not strictly Gaussian, since the Gabor-filter output is never negative. Possibly, the Rayleigh-Rice distribution would be more appropriate, but then the analysis becomes more complex with little impact.)

Given two textured regions A and B and a Gabor filter G_f (3.1), let the output of G_f be represented by the probability density function p_A when G_f is applied to A and by p_B when G_f is applied to B . Consider the following experiment: apply G_f to a textured region (either A or B) and record the output m , at some random position (x, y) , in the random variable z . The problem is to decide whether the random sample was taken from region A (hypothesis H_0) or from region B (hypothesis H_1). Define a decision point d , such that if $z < d$, then the sample is presumed to be from region A (accept hypothesis H_0). Otherwise, it is presumed to be from B (accept hypothesis H_1). For this experiment there are two possible errors; accepting H_1 when H_0 is true (Type I error), or accepting H_0 when H_1 is true (Type II error). The goal is to minimize the sum of these two error probabilities.

The solution to this problem is well known (e.g., see [76]), and reduces to finding the decision point d such that the likelihood ratio $\lambda(z) \triangleq p_B(z)/p_A(z)$ satisfies

$$\lambda(d) = P_0/(1 - P_0) \quad (7.4)$$

where P_0 is the prior probability that the region is A . If we assume that the two regions have the same area and are equally likely to occur, then $P_0 = 1 - P_0 = 1/2$, and $\lambda(d) = 1$. Thus the problem reduces to finding d such that $p_A(d) = p_B(d)$.

Let the normal distribution functions p_A and p_B have parameters (μ_A, σ_A) and (μ_B, σ_B) respectively, and without loss in generality, assume $\mu_A \leq \mu_B$. Then, equating

$p_A(d)$ to $p_B(d)$ gives

$$\frac{1}{\sigma_A \sqrt{2\pi}} \exp \left[\frac{-(d - \mu_A)^2}{2\sigma_A^2} \right] = \frac{1}{\sigma_B \sqrt{2\pi}} \exp \left[\frac{-(d - \mu_B)^2}{2\sigma_B^2} \right] \quad (7.5)$$

Solving for d produces

$$d = \frac{(\mu_A \sigma_B^2 - \mu_B \sigma_A^2) \pm \sigma_A \sigma_B Z}{\sigma_B^2 - \sigma_A^2} \quad (7.6)$$

where

$$Z = \sqrt{(\mu_B - \mu_A)^2 + 2(\sigma_B^2 - \sigma_A^2) \ln(\sigma_B/\sigma_A)} \quad (7.7)$$

and d is chosen such that $\mu_A \leq d \leq \mu_B$.

The error probabilities can then be computed as

$$P_I = \frac{1}{\sigma_A \sqrt{2\pi}} \int_d^\infty \exp \left[\frac{-(d - \mu_A)^2}{2\sigma_A^2} \right] \quad (7.8)$$

$$P_{II} = \frac{1}{\sigma_B \sqrt{2\pi}} \int_{-\infty}^d \exp \left[\frac{-(d - \mu_B)^2}{2\sigma_B^2} \right] \quad (7.9)$$

where P_I and P_{II} are the Type I and Type II error probabilities. Then the total error probability P_E becomes $P_E = P_I + P_{II}$. As P_E becomes small, the probability of mistaking one region for the other becomes small. Thus P_E is a reasonable indicator of step-signature quality. Per step 2 of the GFFS algorithm, P_E is computed for each Gabor-filter center frequency (U, V) of interest. Thus P_E depends on (U, V) ; i.e.,

$$P_E(U, V) = P_I(U, V) + P_{II}(U, V) \quad (7.10)$$

If the parameters of p_A and p_B are known, it is a simple matter to compute P_E .

In practice, however, only estimates of these parameters are available. Estimates of (μ_A, μ_B) can be obtained from the sample means $(\bar{\mu}_A, \bar{\mu}_B)$, and estimates of (σ_A^2, σ_B^2) can be obtained from the sample variances $(\bar{\sigma}_A^2, \bar{\sigma}_B^2)$. The sample mean and the sample variance are both unbiased maximum likelihood estimators. Thus, as the number of available samples approaches infinity, the error in estimating P_E using the sample means and variances approaches zero.

7.2.2 Gabor-Filter Application via Windowed Fourier Transforms

Step 1.b of the GFFS algorithm requires the application of multiple Gabor filters (one filter for each 2-D frequency to be tested) to the randomly selected points within each of the two textured regions. An efficient method for performing this operation is based on the windowed Fourier transform [77].

The windowed Fourier transform is similar to the classic Fourier transform except that the input is premultiplied by a window function. To compute the windowed Fourier transform F , the following equation is evaluated:

$$F_{X,Y}(U, V) = \int \int i(x, y) w(x - X, y - Y) \exp[-j(Ux + Vy)] dx dy \quad (7.11)$$

(all integrals range from $-\infty$ to ∞ unless otherwise stated). Here, w is the window function, i is the image to be transformed, and F is a function of frequency (U, V) , and window position (X, Y) .

The parallel application of a family of Gabor filters to an image at a point is a special case of applying a windowed Fourier transform at the point in the image [78, 79].

To show this, let p be the result of convolving an image i with a GEF h . Then

$$\begin{aligned} p(x, y) &= h(x, y) * i(x, y) \\ &= \int \int i(\alpha, \beta)h(x - \alpha, y - \beta)d\alpha d\beta \end{aligned}$$

Consider one specific point in the convolution (X, Y) . Then

$$\begin{aligned} p(X, Y) &= \int \int i(\alpha, \beta)h(X - \alpha, Y - \beta)d\alpha d\beta \\ &= \int \int i(\alpha, \beta)g((X - \alpha)', (Y - \beta)')\exp[j(U(X - \alpha) + V(Y - \beta))]d\alpha d\beta \end{aligned}$$

where $[(x - \alpha)', (y - \beta)']$ refer to rotated spatial coordinates as defined in Chapter 3. After rearranging terms and factoring out the constant complex exponential $K = \exp[j(UX + VY)]$, we have

$$p(X, Y) = K \int \int i(\alpha, \beta)g((X - \alpha)', (Y - \beta)')\exp[-j(U\alpha + V\beta)]d\alpha d\beta \quad (7.12)$$

Defining the window function w in (7.11) as $w(x, y) = g(-x', -y')$ gives

$$p(X, Y) = K \int \int i(\alpha, \beta)w(\alpha - X, \beta - Y)\exp[-j(U\alpha + V\beta)]d\alpha d\beta \quad (7.13)$$

Observe that, except for the constant K , equations (7.11) and (7.13) are equivalent.

Computing the complex magnitude of (7.13) eliminates K (which represents a constant

phase shift), resulting in

$$|p(X, Y)| = \left| \int \int i(\alpha, \beta) w(\alpha - X, \beta - Y) \exp[-j(U\alpha + V\beta)] d\alpha d\beta \right| \quad (7.14)$$

This justifies (7.1).

The previous development was based on continuous functions. Thus, X, Y, U, V represent continuous variables. These arguments can be easily extended to the discrete case, where X, Y, U, V take on discrete values. In the discrete case, the windowed Fourier transform is implemented using the DFT. Then, (X, Y) refer to image pixels, and (U, V) refer to the DFT frequencies. Thus, if an image is multiplied by a truncated Gaussian centered at image point (X, Y) , and the DFT magnitude is computed, this approximates the application of a family of Gabor filters to the image at the point (X, Y) , where each filter's center frequency corresponds to one of the DFT frequencies. Thus, computing a single DFT is equivalent to determining the output from a family of Gabor filters at a single point, where the center frequencies of the filters span the frequency domain of the image. It should be noted that a Gabor filter could be designed with a center frequency other than one of the DFT frequencies. Thus, GFFS does not apply all possible Gabor filters to an image. Later in this section (under "other issues"), arguments will be presented suggesting that these omissions are not significant.

7.2.3 Specifying σ_x and σ_y

This section examines some heuristics for specifying σ_x and σ_y . It is assumed that, in most cases, $\sigma_x = \sigma_y = \sigma$; thus, the parameter θ is immaterial. Consider the

formation of a step signature. As σ increases, the S/N increases due to a reduction in the noise component (per (7.6) and (7.10)). This occurs for two reasons. First, as window size increases, the computed value of the windowed Fourier transform (WFT) at a point is determined by a larger neighborhood of image pixels. This causes the WFT output to be less sensitive to window position perturbations, thus reducing output variation (i.e., noise). Secondly, for accurate sampling, the possible window positions are restricted to those that approximately keep the window within the region boundaries (the term approximately is used since the window is a Gaussian with infinite extent and will always extend beyond the region bounds). As window size increases, however, the extent of possible window positions decreases. Since the WFT output now varies slowly with position, reducing the size of the sampling area further reduces the output variation. In the limit as the window size approaches the size of the region, the variation in WFT output goes to zero. This causes the S/N to approach infinity, which suggests that σ should be made as large as possible.

The fallacy in this line of thinking is that if σ is made arbitrarily large, any measure of region variability is lost. It must be remembered that in a real texture-segmentation problem, the region size and boundaries are unknown. In that case, if σ is too large, the window can significantly overlap regions, thus reducing discriminability. Therefore, the choice of σ must be guided by practical considerations. Section 6.1 showed that the choice of σ is a tradeoff between discriminability and boundary localization, and that for many strongly-ordered textures, a good compromise is to choose σ approximately equal to the texel spacing. As we will soon see, however, certain textures require a filter configuration that employs a much smaller σ . In practice the GFFS algorithm described

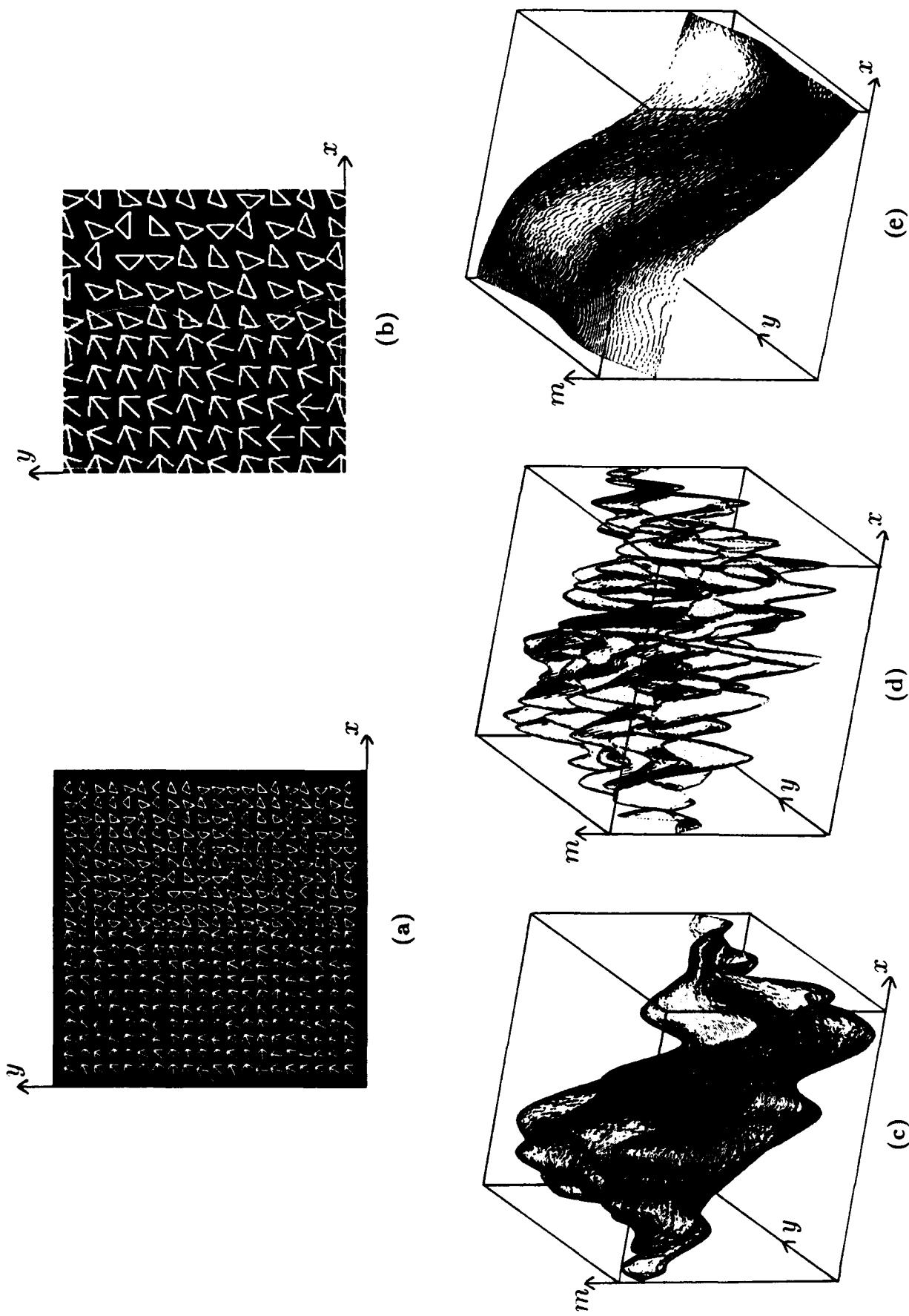
in Section 7.1 produces similar results over a wide range of σ 's. Thus, *it suffices to run the algorithm several times using a few widely spaced values of σ , and compare results.*

7.2.4 Modified Version of Gabor Filter

Fig. 7.1a is an example of a synthetic texture consisting of arrows and triangles. Fig. 7.1c shows the output of a Gabor-filter with $\sigma_x = \sigma_y = \sigma$, and σ equal to the texel spacing. As can be seen in Fig. 7.1c, the step signature is accompanied by considerable variation. This effect is typical of strongly-ordered textures whose texels exhibit pose and/or shape perturbations. The problem is that when σ is large, the bandwidth of the Gabor filter is very narrow, and thus very selective in frequency. In this case, it is *too* selective in frequency. It not only discriminates between the two differently textured regions, but it also detects local frequency variations *within* a region (caused by the random orientations and perturbations of the texels). One possibility is to reduce the size of σ so that the Gabor filter will adequately discriminate between textures without responding to within-texture variations. There is, however, an undesirable side effect of reducing σ . As σ becomes smaller, the spatial resolution of the Gabor filter increases. This increase in spatial resolution causes the Gabor filter to respond to local *spatial* variations within a texture (e.g., the periodic placement of the texels). This effect is illustrated in Fig. 7.1d, where σ was chosen to be one half of that in Fig. 7.1c. It will be argued (under "other issues") that this texture is wide-sense periodic [80]. If this is true and the period corresponds to the texel spacing, then the first two moments of the texture's graylevel distribution are periodic (i.e., $\mu(r) = \mu(r + T)$

Fig. 7.1. Contrasting outputs from C1 and C2 configuration filters:

- (a) A strongly-ordered synthetic texture pair consisting of arrows and triangles.
- (b) Canny edge detector applied to (a) and superimposed on (a).
- (c) Gabor-filter output using a C1 filter, $\sigma = 24$ pixels, $F = 0.061$ cycles/pixel, $\phi = 45.0^\circ$, $P_E = 0.023$. Maximum & minimum values for $m(x, y)$: 2.72×10^5 , 1.65×10^5 .
- (d) Gabor-filter output using a C2 filter (first-stage output $m(x, y)$ (3.1)), $\sigma = 12$ pixels, $F = 0.061$ cycles/pixel, $\phi = 45.0^\circ$, $P_E = 0.453$. Maximum & minimum values for $m(x, y)$: 2.66×10^5 , 5.57×10^4 .
- (e) the second-stage output $m'(x, y)$ (7.15), $\sigma = 36$ pixels. Maximum & minimum values for $m(x, y)$: 7.40×10^9 , 5.49×10^9 .



and $K(r_1, r_2) = K(r_1 + T, r_2) = K(r_1, r_2 + T)$, where μ is the mean, K is the autocovariance matrix, and the r 's are 2-D position vectors. T is any of three constant 2-D vectors $(x_0, 0)^T, (0, y_0)^T, (x_0, y_0)^T$, where x_0 and y_0 represent the texel periods in x and y). Thus it is not unreasonable to expect that the local spatial-frequency composition and, hence, the Gabor-filter output will also be approximately periodic (in a stochastic sense). Thus the local spatial average of the Gabor-filter output within a textured region should be approximately constant. This spatial average can be computed by applying a Gaussian to the Gabor-filter output as shown below.

$$m'(x, y) = m(x, y) * g'(x, y) \quad (7.15)$$

where $g'(x, y)$ is a Gaussian similar to (3.3). The result of applying this Gaussian is shown in Fig. 7.1e. Note the improvement in signature quality.

Although the GFFS algorithm cannot directly predict the quality of the step signature in Fig. 7.1e due to the two-stage process, GFFS can still be employed successfully. By using small values of σ , we can still determine the best step signature for the first stage. Since the second stage is simply a smoothing operation, the resulting output should still represent the highest quality step signature (for that particular value of σ). The application of a single Gabor filter will be referred to as a *C1 configuration* and the two-stage configuration (Gabor-filter (3.1) followed by a Gaussian smoothing (7.15)) will be called a *C2 configuration*.

In most cases $\sigma_x = \sigma_y = \sigma$ is a reasonable choice. Chapter 5 showed, however, that for strongly-ordered textures, when the texel lattice is not square, an asymmetric

filter is preferable. In that case, the ratio of σ_x to σ_y should be adjusted to match the aspect ratio of the texel-spacing lattice, and θ should be chosen to match the orientation of the lattice.

7.2.5 Other Issues

This section presents other implementational issues for the GFFS algorithm. Although theoretically the Gaussian window used to compute the windowed Fourier transform has infinite spatial extent, in practice it is truncated to some finite window size W (typically 6σ). The number of frequency terms computed by the windowed Fourier transform is determined by the window size W . For consistency it is desirable to compute the same number of frequency components for all window sizes. To achieve this, the windowed data are zero padded to extend it to the size of the full image. Although zero padding does not improve frequency resolution (resolution depends on the size of the window), it does increase the number of frequency components generated. The effect is to provide interpolated frequency terms [77]. A positive side effect of this increase is that the number of Gabor-filter center frequencies (U, V) that are tested is increased.

As mentioned earlier, the GFFS algorithm does not compare all possible center frequencies (a formidable task). There are two reasons why this is not necessary. First, Appendix B shows that for strongly-ordered textures, the greatest difference in step height occurs when the Gabor filter is tuned to a multiple of the reciprocal of the texel spacing (i.e., the frequency of occurrence of the texels). These frequencies, among others, are examined by this method. Secondly, experience has shown that $P_E(U, V)$ (and thus signature quality) degrades gracefully with changes in Gabor-filter center frequency. By

comparing $P_E(U, V)$ at frequencies adjacent to the "best" frequency, it can be verified that that $P_E(U, V)$ degrades gracefully in each case.

The random selection of sample points, required in step 1 of the GFFS algorithm, will now be discussed. In many cases, texture can be modeled as a random process. When we are given a sample of a textured region, we are sampling only one instance of this process. In general, this is insufficient for estimating the statistics of the process. In effect, we are finding the "best" filter for one particular instance of the texture. The same filter might be totally ineffective for some other instance. Often, the underlying process is *wide-sense periodic* [80]. Then the process statistics are unique only within a fixed period (e.g., between two adjacent texels). Since a given texture instance will typically contain many periods, sampling a single instance can provide a representative sample. Although not all textures can be considered wide-sense periodic, it is probably a reasonable assumption for strongly-ordered (e.g., Fig. 2.6a and 7.1a) and many disordered textures (e.g., Fig. 2.6c).

The number of points required for a representative sample depends on the variability within a texture. For all examples used in this study, stable results were achieved by taking 200 samples from each region. In general, the number of required samples will be proportional to the region size (in this case, $\approx 0.1\%$ of the region size). Since the region size is $O(N^2)$ and the time complexity of the DFT is $O(N \log N)$, the total time complexity of the algorithm is $O(N^3 \log N)$ (where N is the row or column dimension of the region). In practice, run times are in the neighborhood of 2 hours on a Sun SPARCstation 1.

7.3 Previously Proposed Methods for Designing Gabor Filters

Several other techniques for determining Gabor-filter parameters have previously been suggested. One popular technique is to use heuristics based on neurophysiological and psychophysical studies of the human visual system (HVS) to design a set of filters [56, 60, 74]. (Note that Malik and Perona [56] did not use Gabor filters. They did, however, use functions that are similar.) While this technique has been used effectively to test prototype texture-segmentation schemes, it is a brute force approach providing little insight into the relationship between algorithm output and the filter characteristics that produced the output. Thus, it is difficult to predict how these schemes will perform over a wide range of textures.

A method suggested by the works of Kroese [15] and Fogel and Sagi [55] involves comparing the spectral composition of prototype texels from the regions to be segmented. For uniform textures, Chapter 5 showed that the formation of a step signature is directly related to the difference in frequency content between texels. This method, however, has two limitations. First, it is restricted to strongly-ordered textures. Second, as texel spacing decreases, the texels begin to interact and lose their individual identity. When this occurs, the method becomes ineffective.

Before the GFFS algorithm, the most effective technique for determining Gabor-filter parameters was based on computing the DFT of each textured region [53, 68]. The 2-D frequency component that differs most between regions is then selected. This method will be referred to as the DFT method. The DFT method is equivalent to applying a windowed Fourier transform to a region, where the window is rectangular

and equal to the size of the region. Experimental results indicate that the best choice of center frequency is typically insensitive to window size. Since a rectangular window can be sized to approximate the spatial extent of a Gaussian, the results of the DFT method sometimes predict the same "best" center frequency as GFFS. Although the DFT method is somewhat faster than GFFS, it does not always predict useful center frequencies. Chapter 8 compares the GFFS algorithm with the DFT method for finding filter frequency parameters and describes other limitations of the DFT method.

7.4 Determining Filter Parameters for Other Signature Types

The GFFS algorithm presented in Section 7.1 assumes that to distinguish between two textured regions, a step signature is desired. The algorithm can be easily modified to find Gabor filters that generate valley/ridge and difference-in-variance signatures. For the valley/ridge, samples are pooled from both textured regions *A* and *B* (for this type of signature, regions *A* and *B* are typically identical) this forms sample set *A*. Samples are then collected along the texture boundary to give sample set *B*. The rest of the algorithm does not change. With this modification, the algorithm will find the center frequency that produces the highest ridge or deepest valley.

To determine Gabor filters that will produce the largest difference in output variance, simply compare the differences in sample variances and choose the DFT frequency that generates the largest difference. It is also necessary to check that the means are similar, as this is an important consideration for subsequently transforming the difference in variance to a difference in mean (Section 5.2.3). The question of which type of signature is most appropriate for distinguishing a given texture pair remains open.

Chapter 8

Results

This chapter presents 1-D and 2-D experimental results corroborating the analysis done in previous chapters.

8.1 1-D Results

Fig. 8.1 gives examples of the 1-D textures used in the analytical work of Chapter 4, and Fig. 8.2 gives a plot of four filter outputs produced by applying a Gabor filter to 1-D textured images.

Each image consists of two regions 1 and 2, consisting of six texels. Region 1 is to the left of zero and region 2 is to the right. The texels in region 1 are defined by (4.1) and the texels in region 2 are defined by (4.2). All texels are spaced 24 units apart and are 16 units wide (i.e., $\Delta x = 16$). The texture frequencies for regions 1 and 2 are ω_1 and ω_2 respectively, and the phase difference $(\phi_1 - \phi_2)$ between regions is ϕ . By adjusting these parameters, discontinuities in frequency, phase, or both can be induced between regions.

Curve A is the result of a difference in texture frequency between regions. The Gabor filter used to produce this curve has a center frequency that matches the texture frequency of region 1. Thus, the Gabor-filter output is greater for region 1 than for region 2. As the figure shows, the filter output approximates a step. Note that the position of the texture boundary ($x = 0$) is located approximately at the middle of the step. Curve

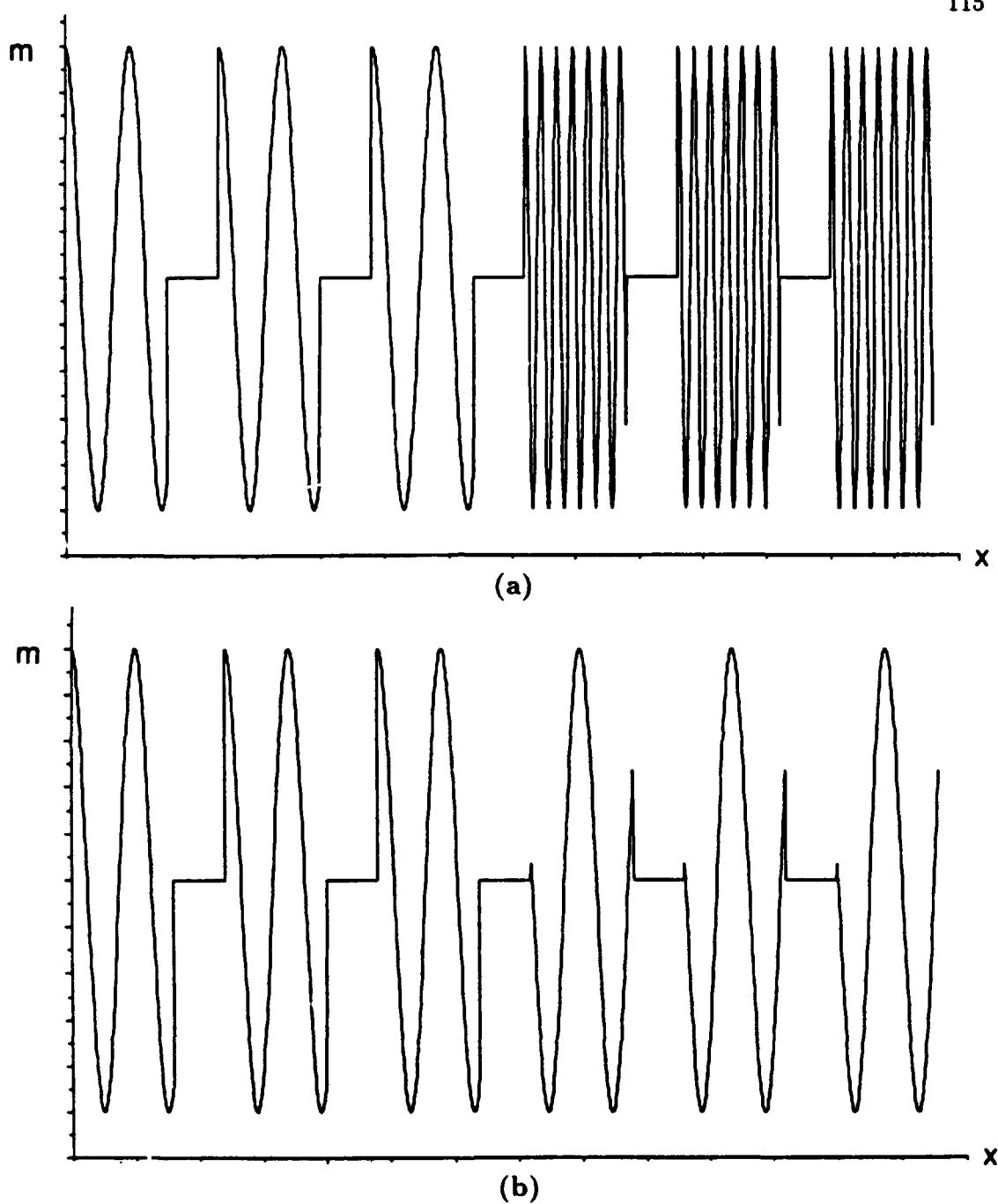


Fig. 8.1. Examples of 1-D textures complying to the 1-D texture model of Section 4.1.

(a) 1-D texture constructed from the model (4.3), where the texture frequencies differ between regions (the frequency differences have been exaggerated for clarity).

(b) 1-D texture constructed from the model (4.3), where the texture elements differ in phase between regions (the frequency has been reduced for clarity).

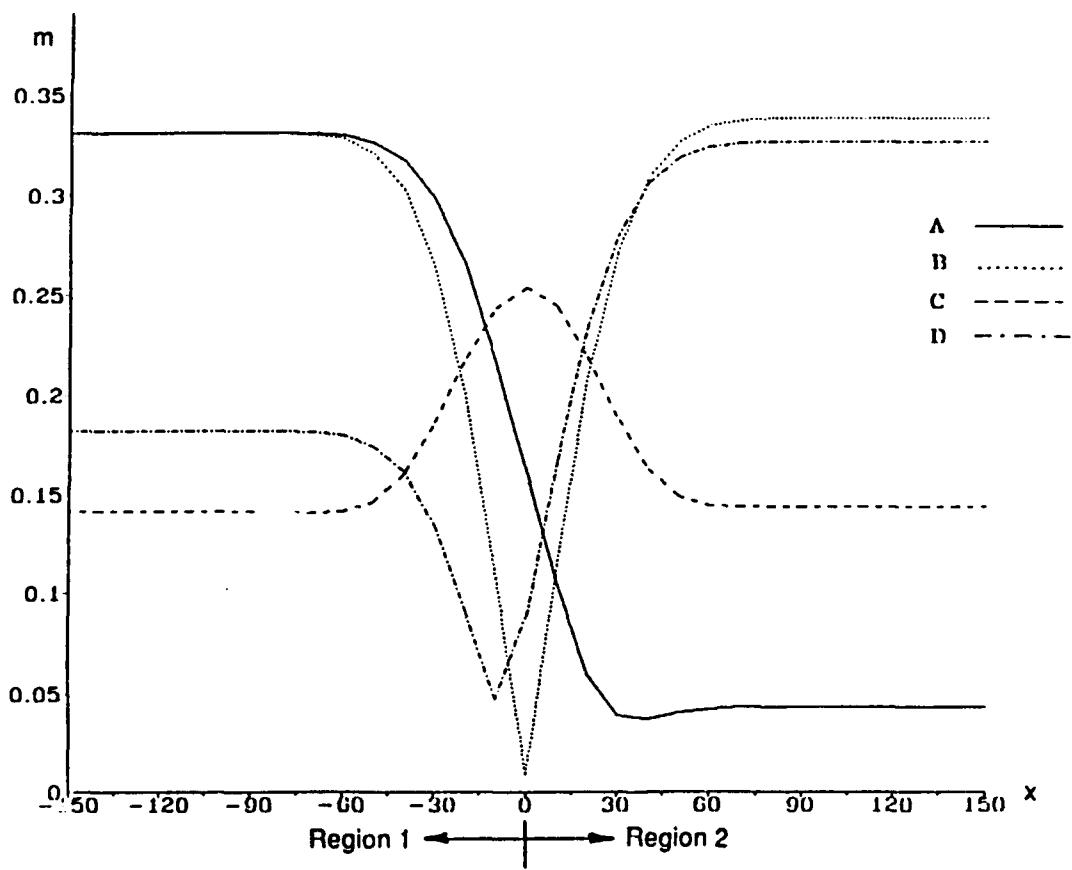


Fig. 8.2. 1-D Gabor-filter outputs m derived from 1-D textures (define in (4.1) and (4.2)). ω_1 and ω_2 are the texture frequencies in radians per unit distance for regions 1 and 2 respectively, and ϕ is the phase difference between regions in radians. ω_c is the filter center frequency in radians per unit distance. The location $x = 0$ represents the texture boundary.

Curve A: $\omega_1 = 10.47$, $\omega_2 = 9.43$, $\omega_c = 10.47$, $\phi = 0$ (differing texture frequency).

Curve B: $\omega_1 = 10.47$, $\omega_2 = 10.47$, $\omega_c = 10.47$, $\phi = \pi$ (textures of same texture frequency, but out of phase).

Curve C: $\omega_1 = 5.03$, $\omega_2 = 5.03$, $\omega_c = 5.03$, $\phi = 1.57$ (textures of same texture frequency, but out of phase).

Curve D: $\omega_1 = 10.24$, $\omega_2 = 10.47$, $\omega_c = 10.47$, $\phi = 1.5$ (differences in texture frequency and phase).

B is the result of a difference in phase. Note that the texture frequency is the same for both textures, but there is a π radian phase difference between them. Also, note that the texture frequencies equal the center frequency of the filter. Observe that the curve forms a valley as expected from the analytical work. In this case, the position of the boundary occurs at the global minimum of the Gabor-filter output. Curve C also represents a difference in phase, but a ridge is formed in the output rather than a valley. The 1-D analysis, however, does not reveal how a ridge can occur at a phase discontinuity. Note that the texture boundary occurs at the global maximum (minimum) of the Gabor filter output for curve C (B). Curve D is produced by a combination of frequency and phase changes. In this case the output is in-between a step and a valley. Depending on the frequency and phase values, either the step or the ridge/valley will dominate for such cases. For curve D, the texture boundary is neither at the minimum nor at the middle of the step. Thus, there exists an inherent degree of boundary uncertainty with this profile, as is the case with certain perceived texture boundaries. For all filters used in Fig. 8.2, σ equals the texel spacing.

8.2 2-D Results

All images used in the examples to follow consist of two regions. Except for the examples of natural textures, each region is composed of a collection of synthesized texels. Each texel is formed from line segments 20 pixels long by 2 pixels wide. The average intensity difference between regions is minimized by using approximately the same number of pixels in each texel. The size of the images is 512 \times 512 pixels.

Except where noted, the Gabor-filter parameters were determined as follows. Section 6.1 recommended that $\sigma_x = \Delta x$ and $\sigma_y = \Delta y$. For most of the examples, $\Delta x = \Delta y$. Hence, $\sigma_x = \sigma_y \triangleq \sigma = 24$ pixels. The aspect ratio $\lambda = 1$. The GFFS algorithm developed in Chapter 7 was used to determine the center frequency (U, V) . This algorithm finds the harmonic $(U = 2\pi\hat{k}/\Delta x, V = 2\pi\hat{l}/\Delta y)$ that produces the largest step (or the deepest valley, or the highest ridge for texture-phase differences). In the figure captions, the Gabor-filter center frequencies (U, V) are reported in polar coordinates (F, ϕ) so that the orientation of the filter's sinusoid is explicit ($F = \Omega/2\pi$ in (3.6)).

The input images are defined digitally. Thus, aliasing in the images is not a issue. Aliasing is an issue, however, for the GEF, since it must be sampled before applying it to the image. Since the GEFs are not bandlimited, some aliasing will occur regardless of the sample rate. Bovik *et al.* derived the required sample rate for various percentages of alias energy [53]. In the examples, the GEFs are sampled so that the energy due to aliasing is, in most cases, $< 1\%$.¹ Exceptions occur for the filter outputs shown in Figs. 8.11c and 8.12d, where the aliasing energy is 7.4% and 12% respectively. The increase in aliasing is due to the high center frequencies used in these filters, but this does not pose a problem. The GFFS algorithm still finds the most discriminating digitized filter, even though it might be somewhat distorted.

Since the GEFs are not spatially limited, truncation is necessary. The GEFs are truncated to a width of 6σ , which represents an error of about 0.2%. Except in Fig. 8.13, all points within 1/2 the filter width from the boundary are discarded in the

¹Bovik *et al.*'s calculation is conservative due to using $\ln 2$ instead of $\sqrt{\ln 2}$ in their equation for γ_B [53].

final output to eliminate the wraparound error that arises in discrete convolution. For this reason some of the output figures appear truncated. Some examples show the results of applying a Canny edge detector to a filtered image. This gives possible subsequent segmentations. No effort was made, however, to optimize this detector or to optimize the segmentation algorithm applied to the filtered image. Appendix D discusses the implementation details of filter application and edge detection.

8.2.1 Difference in Texel Type

Fig. 8.3a illustrates a uniformly textured image consisting of +s and Ls. Fig. 8.3b gives a plot of a Gabor-filter output $m(x, y)$ versus x and y . The vertical axis gives $m(x, y)$ (the maximum and minimum filter outputs are indicated on the axis), and the two axes approximately horizontal and into the page represent x and y . All Gabor-filter outputs are depicted this way.

The shape of the profile is predominantly a step function with some undershoot present. The output of a Canny edge detector [75] applied to the Gabor-filtered image is shown superimposed on the original image - see Fig. 8.3c. As Figs. 8.3b and 8.3c indicate, the boundary between the two textured regions is well localized.

An estimate of step height in Fig. 8.3b can be found by using the equations for A_1 (5.22) and A_2 (5.23). These equations imply that the ratio $|T_1|/|T_2|$ is a relative measure of the step height. Letting T_1 correspond to a "+" and T_2 correspond to an "L" for the image in Fig. 8.3a, gives $T_1 = 22.38$ and $T_2 = -1$. Thus, the predicted step height is $|T_1|/|T_2| = |22.38|/|-1| = 22.38$. For Fig. 8.3b, the ratio of left-region and right-region heights is 22.70, which is in good agreement with the predicted value.

Fig. 8.3. Uniformly textured image consisting of +s and Ls. $\Delta x = \Delta y = 24$ pixels. Part b is a 3-D plot of $m(x, y)$ versus x and y . The vertical axis gives m . The axes that are approximately horizontal and into the page represent the x and y axes respectively:

- (a) Input image.
- (b) Gabor-filter output m exhibiting a step signature. Filter parameters: $F = 0.059$ cycles/pixel, $\phi = 135^\circ$, and $\sigma = 24$ pixels. Maximum & minimum values for $m(x, y)$: 2.0×10^5 , 1.0×10^2 .
- (c) Superposition of the input image and the output (vertical white line) of a Canny edge detector applied to (b).
- (d) Gabor-filter output m exhibiting overshoot. Filter parameters: $F = 0.010$ cycles/pixel, $\phi = 135^\circ$, and $\sigma = 24$ pixels. Maximum & minimum values for $m(x, y)$: 8.9×10^3 , 1.1×10^1 .

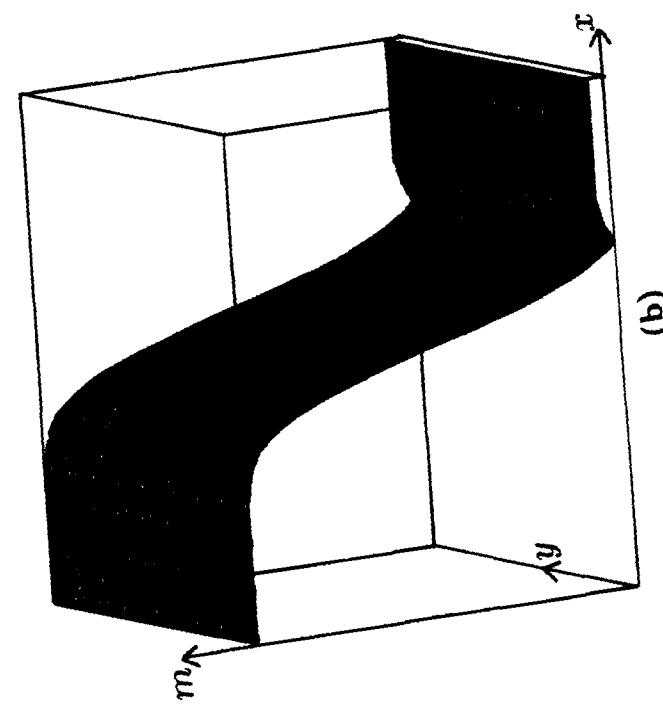
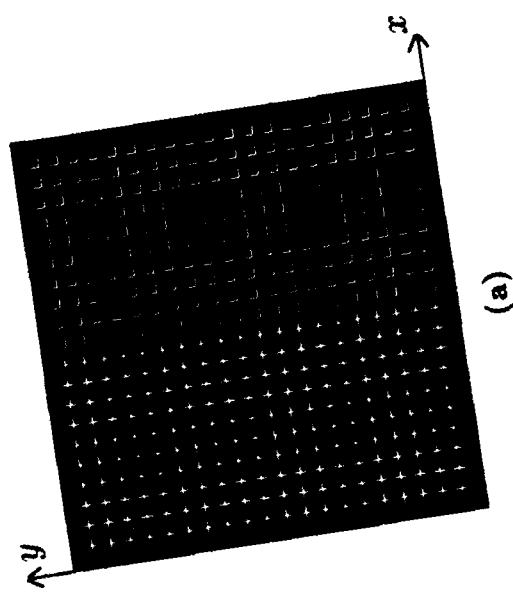
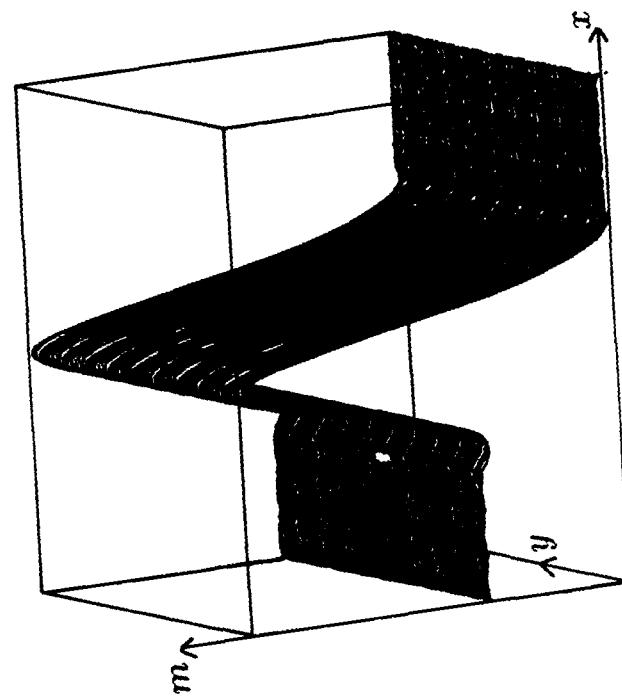
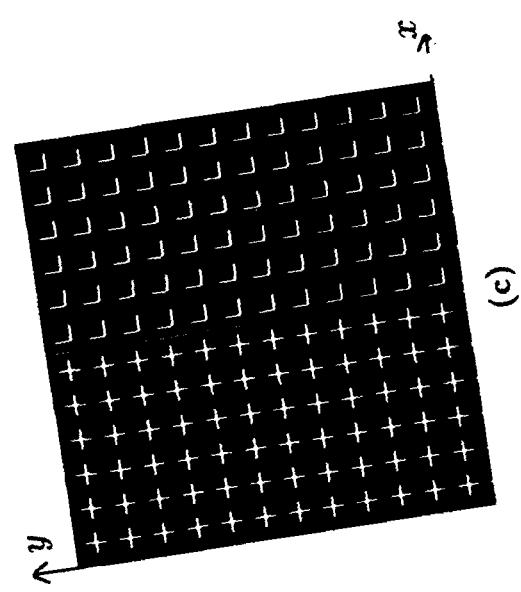


Table 8.1. Comparison of actual and predicted Gabor-filter output values for the step signature in Fig. 8.3b.

location	actual	predicted
+ region	20.0	22.4
max. undershoot	0.010	0.024
L region	0.88	1.00

In this example, undershoot occurs in the Gabor-filter output, because T_2 is negative. This phenomena is discussed in Section 5.2.1 and in Appendix B. Although (5.21) demonstrates the possibility of undershoot, (5.18) must be evaluated to determine the position and extent of undershoot. Letting $T_1 = 22.38$ and $T_2 = -1$ in (5.18), an estimate of the signature in Fig. 8.3b was computed. Table 8.1 compares actual and predicted signature values at selected positions. (The actual values have been scaled by a constant factor for comparison.) This example illustrates that even if the Gabor-filter center frequency equals an harmonic, undershoot can occur; however, overshoot cannot occur for this situation, as discussed in Appendix B.

Both overshoot and undershoot can occur if the Gabor filter is not tuned to an harmonic. Fig. 8.3d gives an example. This figure represents the output of a Gabor filter having center frequency ($U = 0.0283, V = -0.0283$) (input is Fig. 8.3a). The texel spacing is 24 pixels. Thus, the closest harmonic to the filter center frequency is $\hat{k}/\Delta x = \hat{l}/\Delta y = 1/24 = 0.0417$. The center frequency then is displaced $\delta U = \delta V = 0.0134$ cycles/pixel away from the nearest harmonic. Using these parameters in (B.1) to compute predicted values of Gabor-filter output $m(x, \cdot)$ at selected x reveals good

Table 8.2. Comparison of actual and predicted Gabor-filter output values for the step signature in Fig. 8.3d.

location	actual	predicted
+ region	33.5	38.2
max. overshoot	88.7	85.1
max. undershoot	0.11	0.80
L region	1.35	1.71

agreement with the actual values of Fig. 8.3d (see Table 8.2).

8.2.2 Difference in Texel Orientation

Fig. 8.4a shows a uniformly textured image consisting of texels that differ in orientation. Figs. 8.4b and 8.4c show the outputs of two Gabor filters that use the same center frequency but use different values for σ . In Fig. 8.4b, σ equals the texel spacing, and a smooth step signature is achieved. The region on the right produces the greatest Gabor-filter output m , because the orientation of the Gabor-filter sinusoid matches the texel orientations on the right. In Fig. 8.4c, $\sigma = 8$, which is 1/3 of the texel spacing. For this small σ , the GEF doesn't cover multiple texels as it moves across the image, resulting in ripple in the filter output. The resulting signature, though, exhibits a sharper step transition than Fig. 8.4b.

8.2.3 Differences in Horizontal and Vertical Texel Spacing

Fig. 8.5a is equivalent to Fig. 8.3a, with the exception that $\Delta y = 2\Delta x$ for the two textures. Fig. 8.5b shows a corresponding Gabor filter output when its GEF uses

Fig. 8.4. Uniformly textured image consisting of texels that differ in orientation. $\Delta x = \Delta y = 24$ pixels. Common filter parameters are $F = 0.059$ cycles/pixel, $\phi = 135^\circ$:

- (a) Input image.
- (b) Gabor-filter output m exhibiting a smooth step signature: $\sigma = 24$ pixels. Maximum & minimum values for $m(x, y)$: 4.2×10^5 , 1.2×10^5 .
- (c) Gabor-filter output m exhibiting a step with ripple: $\sigma = 8$ pixels. Maximum & minimum values for $m(x, y)$: 3.1×10^4 , 4.5×10^3 .

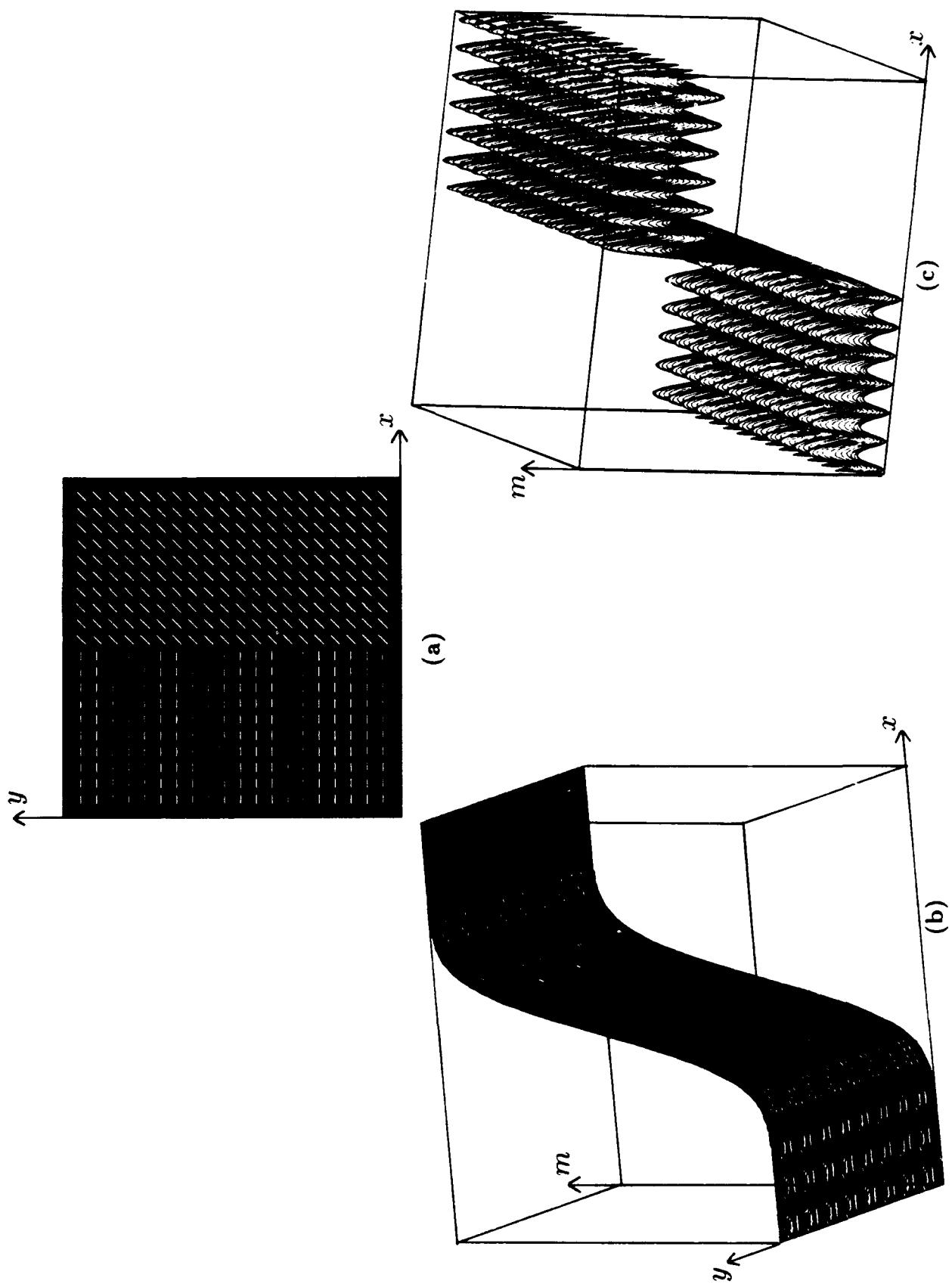
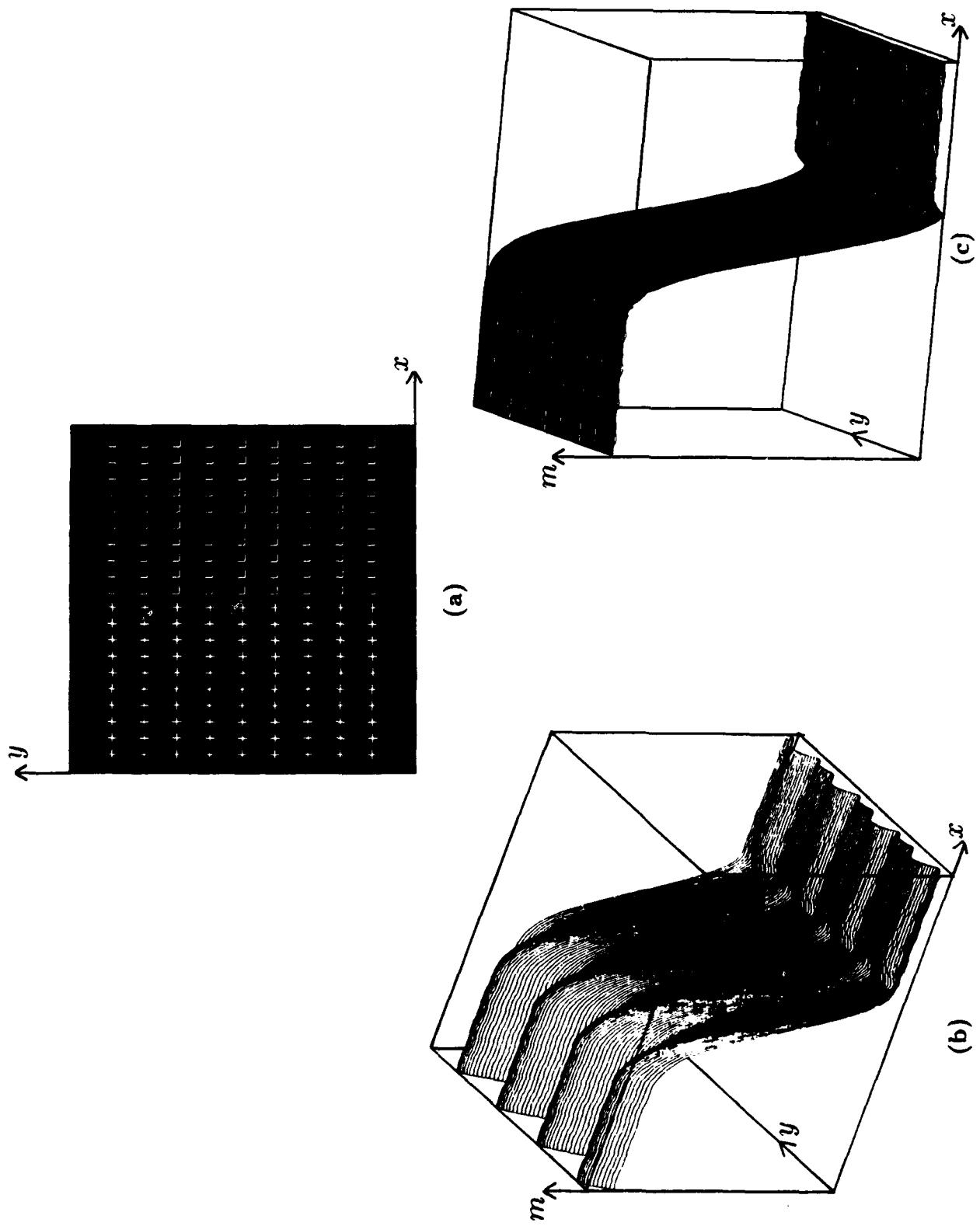


Fig. 8.5. Uniformly textured image with a different texel spacing in the x and y directions. $\Delta x = 24$, $\Delta y = 48$ pixels. Common filter parameters are $F = 0.059$ cycles/pixel, $\phi = 135^\circ$:

- (a) Input image.
- (b) Gabor-filter output m with a circularly symmetric GEF. Filter parameters: $\sigma = 18$ pixels, and $\lambda = 1$. Maximum & minimum values for $m(x, y)$: 1.0×10^5 , 3.1×10^2 .
- (c) Gabor-filter output m with asymmetric GEF. Filter parameters: $\sigma = 18$ pixels, and $\lambda = 2$. Maximum & minimum values for $m(x, y)$: 1.1×10^5 , 1.5×10^2 .



a symmetrical Gaussian (i.e., $\sigma_x = \sigma_y$ or $\lambda = 1$). Note the occurrence of significant ripple in the y direction. Fig. 8.5c gives the Gabor-filter output for a filter with $\lambda = 2$ ($\sigma_y = 2\sigma_x$). For this case, the output is “smooth” in both x and y and resembles the smooth step of Fig. 8.3b.

8.2.4 Texture-Phase Differences

Fig. 8.6a consists of two identically textured regions, but the regions are shifted relative to each other in both the x and y directions (Fig. 1.3b gives a simpler example). The texel spacing is 24 pixels in both x and y , and the region on the right is shifted -8 pixels in the y direction and -4 pixels in the x direction relative to the region on the left. Fig. 8.6b shows the output of a Gabor filter that is tuned to a frequency that is close to an harmonic. Note that a valley signature results.

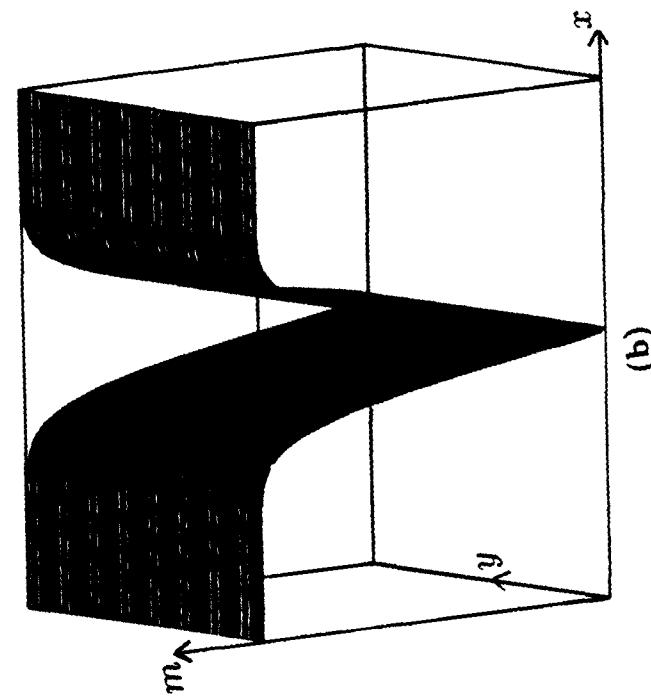
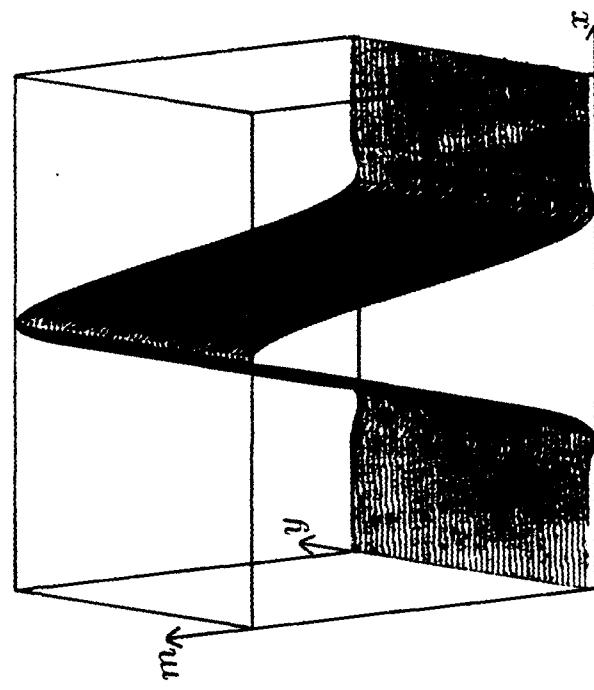
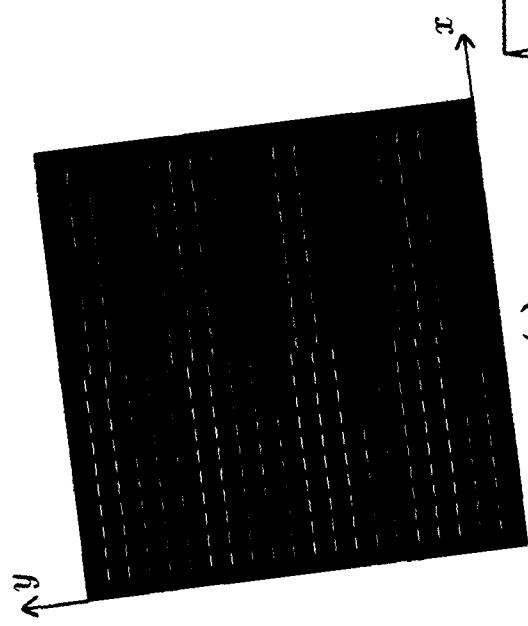
The harmonics are located at $(0.042k, 0.042l)$ and the center frequency of the filter is $(U = 0.0039, V = -0.0410)$; so the indices of the nearest corresponding harmonic are $\hat{k} = 0$ and $\hat{l} = -1$. The filter’s specified center frequency (U, V) differs from one of the harmonics by an amount $(\delta U = 0.039, \delta V = 0.001)$. From the analysis of Section 5.2.2, a ridge-signature output is expected. Using the method developed in Appendix C for computing z , however, reveals that $z = 0.068$. Thus, even though δU is nonzero, a valley is predicted rather than a ridge. This occurs because

$$\psi = 2\pi[\hat{k}\delta x/\Delta x + \hat{l}\delta y/\Delta y] = 2\pi(-1)(-8/24) \text{ radians} \equiv 120^\circ$$

which is significantly different from ψ_{\max} in Table 5.1. The predicted depth of the valley

Fig. 8.6. Uniformly textured image with regions shifted both horizontal and vertical, thus producing a texture-phase discontinuity. $\Delta x = \Delta y = 24$ pixels:

- (a) Input image.
- (b) Gabor-filter output m exhibiting a valley signature. Filter tuned to an harmonic. Filter parameters: $F = 0.0412$ cycles/pixel, $\phi = -84.56^\circ$, and $\sigma = 24$ pixels. Maximum & minimum values for $m(x, y)$: $2.4 \times 10^5, 2.6 \times 10^4$.
- (c) Gabor-filter output m exhibiting a ridge signature. Filter not tuned to an harmonic. Filter parameters: $F = 0.0427$ cycles/pixel, $\phi = 74.05^\circ$, and $\sigma = 24$ pixels. Maximum & minimum values for $m(x, y)$: $1.9 \times 10^5, 5.5 \times 10^4$.



130

then is 0.068 times the Gabor-filter output at points far removed from the transition. The maximum and minimum outputs of Fig. 8.6b imply that the valley is in fact 0.124 times the value at remote points. This error can be explained in part by the sensitivity of (5.42) to ψ . Recall that ψ depends on the texture-phase shift in the image. Reducing the phase shift in the y direction by just 1/2 pixel changes the predicted value to $z = 0.141$.

Again using Fig. 8.6a, a Gabor filter was applied whose center frequency exactly matched the first harmonic in u (output not shown but similar to Fig. 8.6b). For this case, the expression for the relative depth of the valley is derived from (5.29), which is much less sensitive to ψ . Using (5.29) results in $z = 0.5$, which differs by only 7% from the actual valley depth.

Fig. 8.6c shows the result of processing Fig. 8.6a using a Gabor filter that is tuned to a frequency significantly displaced from an harmonic. A ridge signature results. The harmonics are located at $(0.042 \cdot k, 0.042 \cdot l)$ and the center frequency of the filter is $(U = 0.012, V = -0.0410)$, so $\hat{k} = 0$ and $\hat{l} = 1$. σ_x and σ_y again equal 24. From (5.26)

$$\psi = 2\pi[\hat{k}\delta x/\Delta x + \hat{l}\delta y/\Delta y] = 2\pi(1)(-8/24) \text{ radians} \equiv -120^\circ$$

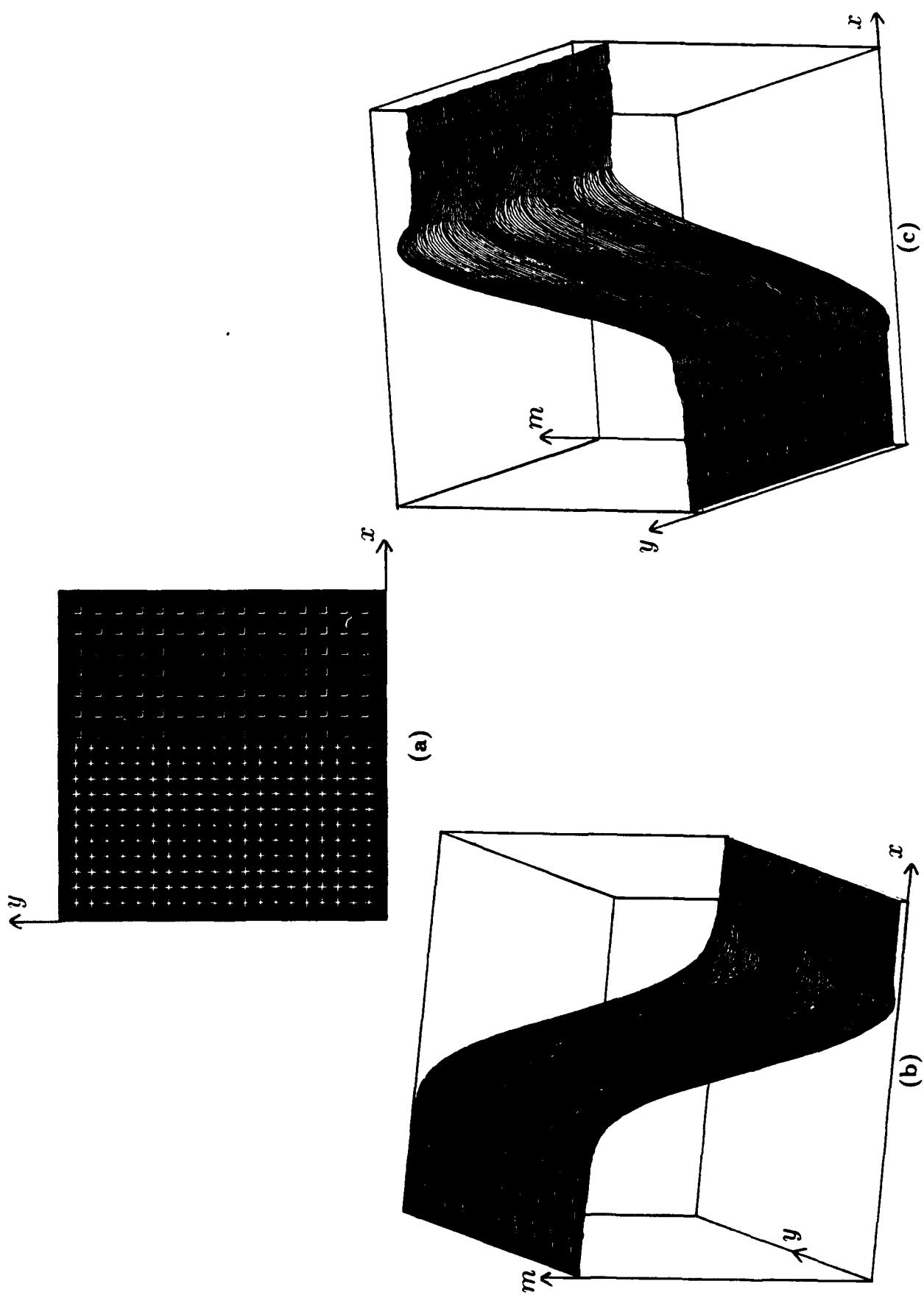
and the method developed in Appendix C predicts a ridge height $z = 2.973$. The actual ridge height in Fig. 8.6c is 3.24, a 9% error.

8.2.5 Texel-Spacing Difference between Regions

Fig. 8.7a shows a uniformly textured image similar to Fig. 8.3a, except that the texel spacing differs between the two regions. Fig. 8.7b shows a corresponding Gabor-

Fig. 8.7. Uniformly textured image with each textured region having a different texel spacing. $\Delta x_1 = \Delta y_1 = 24$ pixels. $\Delta x_2 = \Delta y_2 = 32$ pixels:

- (a) Input image.
- (b) Gabor-filter output m . Filter tuned to an harmonic corresponding to region of $+\mathbf{s}$ (some undershoot present). Filter parameters: $F = 0.059$ cycles/pixel, $\phi = 135^\circ$, and $\sigma = 24$ pixels. Maximum & minimum values for $m(x, y)$: 2.0×10^5 , 5.9×10^1 .
- (c) Gabor-filter output m (some undershoot and overshoot present). Filter tuned to an harmonic corresponding to region of \mathbf{Ls} . Filter parameters: $F = 0.0393$ cycles/pixel, $\phi = 135^\circ$, and $\sigma = 24$ pixels. Maximum & minimum values for $m(x, y)$: 5.8×10^4 , 8.9×10^1 .



filter output, where the filter is tuned to an harmonic corresponding to the region of $+$ s. Although the signature is predominantly a step, some undershoot is present near the texture boundary. Fig. 8.7c shows a similar filter output, but now the filter is tuned to an harmonic for the region of Ls. In this case, both overshoot and undershoot are present. Observe how in each case (Figs. 8.7b and 8.7c) that the region producing the greatest filter response corresponds to the one whose harmonic matches the filter center frequency. The analysis of Appendix B verifies this empirical result.

8.2.6 Nonuniform Textures

Fig. 8.8a depicts a nonuniformly textured image produced by introducing random orientations and positional perturbations into the texels ($+$ s and Ls) of Fig. 8.3a. Fig. 8.8b shows a filter output. The random effects cause large fluctuations in the output. Fig. 8.8c shows the result of applying a Canny edge detector to Fig. 8.8b. Because of the fluctuations, the detected boundary does not perfectly match the “actual” boundary. The predicted boundary is, for the most part, correct to within $\pm 1/2$ texel. For typical nonuniform textures (where the actual texture boundary is not well defined), such fluctuation in the computed texture boundary is expected.

Section 7.2 presented a synthetic texture pair consisting of triangles and arrows, and a corresponding Gabor-filter output signature exhibiting a step signature (Fig. 7.1). This image can also produce a signature exhibiting a step change in average local output variation. Fig. 8.9a shows a Gabor filter output exhibiting such a signature (the input image is Fig. 7.1a). After applying (5.43), the change in average local output variation was transformed to the step signature shown in Fig. 8.9b.

Fig. 8.8. Nonuniformly textured image consisting of +s and Ls. $\Delta x = \Delta y = 24$ pixels:

- (a) Input image.
- (b) Gabor-filter output m exhibiting a “noisy” step signature. Filter parameters are $F = 0.042$ cycles/pixel, $\phi = 0^\circ$, and $\sigma = 24$ pixels. Maximum & minimum values for $m(x, y)$: 3.6×10^5 , 1.8×10^5 .
- (c) Canny edge-detector output superimposed on input image.

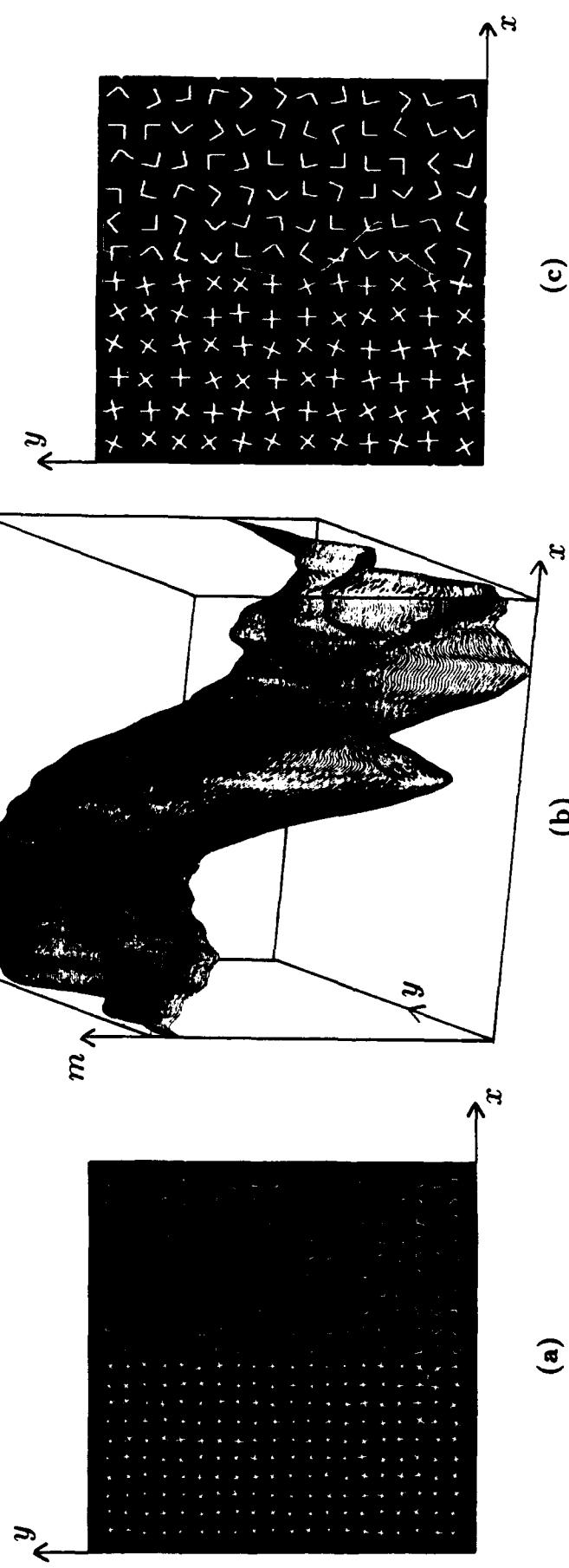
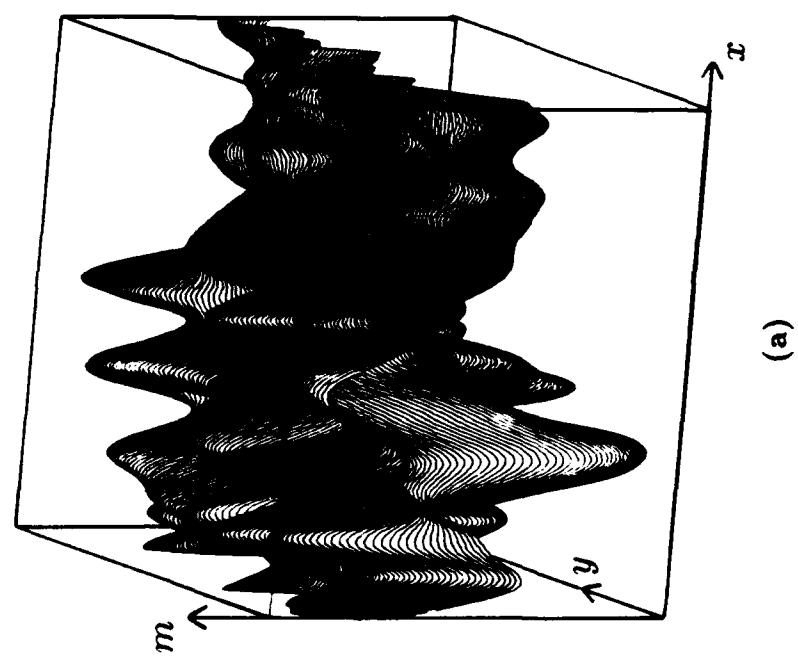
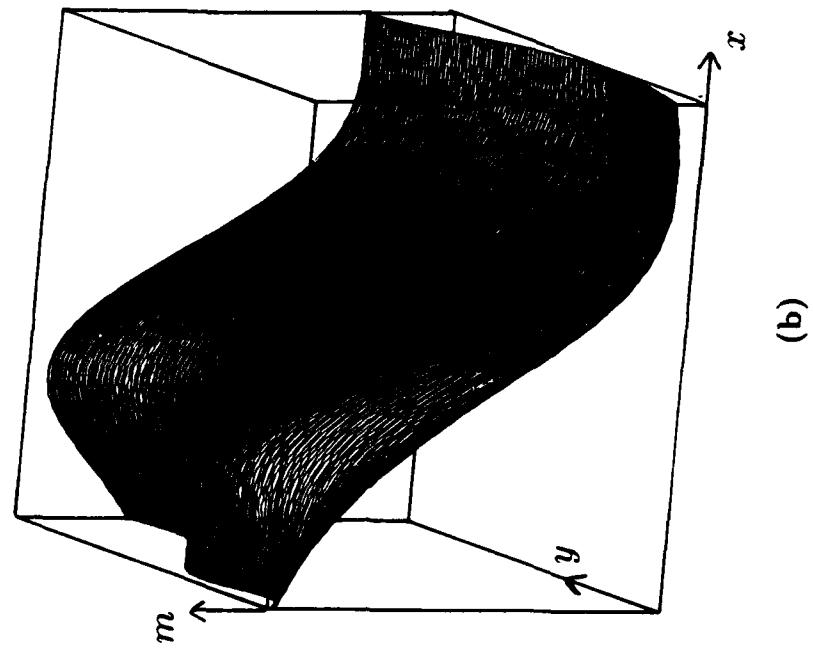


Fig. 8.9. Difference-in-variation signature resulting from arrows-and-triangles image:

(a) Gabor-filter output m exhibiting a step change in average local output variation at the texture boundary. Filter parameters are $F = 0.042$ cycles/pixel, $\phi = 90^\circ$ and $\sigma = 16$ pixels. Maximum & minimum values for $m(x, y)$: 3.1×10^4 , 1.1×10^4 .

(b) Resulting output after (5.43) is applied to (a). Note that the difference in average local output variation in (a) is transformed to a step signature. Maximum & minimum values for $m(x, y)$: 1.5×10^9 , 2.3×10^8 .



As previous examples illustrate, a single stage filter (C1 configuration - Section 7.2) often produces adequate results; however, this is not always the case. The following examples illustrate that signature quality can sometimes be improved by using a C2 configuration filter (see Section 7.2). Frequency parameters were determined by the GFFS algorithm, and results are compared to filters with frequencies determined by the DFT method. Both natural and synthetic textures are examined. All natural textures were digitized (512 x 512 pixels, 256 graylevels) from Brodatz [13], and all texture pairs were adjusted for equal average intensity. For simplicity, only symmetrical Gabor filters were used; i.e., $\sigma_x = \sigma_y = \sigma$. These examples match up textures from various classes (per Rao's classification [12]). This gives a broad, strong test for the validity of the analyses in Chapter 5 and the efficacy of the GFFS algorithm developed in Chapter 6.

Let us begin by examining the triangles-and-arrows image in Fig. 7.1 in more detail. Fig. 7.1c is the output of a Gabor-filter (C1 configuration) with the frequency parameters determined by GFFS and σ equal to the texel spacing. For this example, GFFS predicts similar frequency parameters for σ 's ranging from 12 to 36 pixels. So, in this case, the results are insensitive to σ . Figs. 7.1d and 7.1e show the two stages of a C2 configuration filter applied to Fig. 7.1a. Fig. 7.1b shows the result of applying a Canny edge detector to Fig. 7.1e. The DFT method was also applied to Fig. 7.1a. DFT predicts a similar radial frequency, but the orientation angle ϕ has the opposite sign (-45° as opposed to 45°). The GFFS algorithm ranks the DFT prediction 20th ($\sigma = 24$), with a corresponding P_r value of 0.060. The output signature for the DFT-predicted value (not shown) was examined and found to be similar to Fig. 7.1c.

Fig. 8.10a shows a pair of natural textures. The left region is Brodatz's "grass

lawn" (D9) and the right region is "cotton canvas" (D77). D9 is an example of a disordered texture, while D77 is strongly-ordered [12]. GFFS again predicts the same center frequency for σ ranging from 12 through 36; however, in this case, it agrees with the DFT predicted values. Fig. 8.10c shows the result of applying a C1 configuration filter to Fig. 8.10a. In this case $P_E \ll 0.00001$, which is consistent with the improved signature quality compared to Fig. 7.1c. Figs. 8.10d and 8.10e show the two stages of a C2 configuration filter applied to Fig. 8.10a. For this example, high quality signatures are obtained for both C1 and C2 configurations. Fig. 8.10b shows the result of applying a Canny edge detector to Fig. 8.10e.

Fig. 8.11a consists of "straw matting" (D55) and "raffia" (D84). Both of these textures are strongly-ordered [12]. Notice that although the texels in the two regions are perceptually different, they are similar in size, orientation, and aspect ratio. In this case, for both $\sigma = 36$ and $\sigma = 24$, GFFS predicts two very different frequencies ($F = 0.491, \phi = 59.4^\circ$ and $F = 0.046, \phi = 87.4^\circ$) with similar error probabilities. The center frequency corresponding to the smaller value of F agrees with the DFT predicted value. At $\sigma = 12$, however, GFFS predicts F s that are all much greater than the DFT value. In general, it can be expected that for large σ s, DFT and GFFS will predict similar values. This is because as the effective window size approaches the size of the region, the windowed Fourier transform approaches the discrete Fourier transform of the entire region. Fig. 8.11c shows the result of applying a C1 configuration filter to Fig. 8.11a. Figs. 8.11d and 8.11e show the two stages in applying a C2 filter to the same texture pair. Here we can begin to see a difference in signature quality between the C1 and C2 configurations. Fig. 8.11b shows the result of applying a Canny edge detector

Fig. 8.10. Natural texture pair consisting of Brodatz's "grass lawn" (D9) and "cotton canvas" (D77):
(a) Input image.
(b) Canny edge detector applied to (e) and superimposed on (a).
(c) Gabor-filter output using a C1 filter, $\sigma = 36$ pixels, $F = 0.060$ cycles/pixel, $\phi = 60.8^\circ$, $P_E \ll 0.00001$. Maximum & minimum values for $m(x, y)$: 4.95×10^6 , 4.07×10^3 .
(d) Gabor-filter output $m(x, y)$, $\sigma = 12$ pixels, $F = 0.063$ cycles/pixel, $\phi = 60.0^\circ$, $P_E = 0.002$. Maximum & minimum values for $m(x, y)$: 4.29×10^6 , 2.11×10^3 .
(e) C2 second-stage output $m'(x, y)$ after filtering (d), $\sigma = 36$ pixels. Maximum & minimum values for $m(x, y)$: 1.52×10^{12} , 3.11×10^{11} .

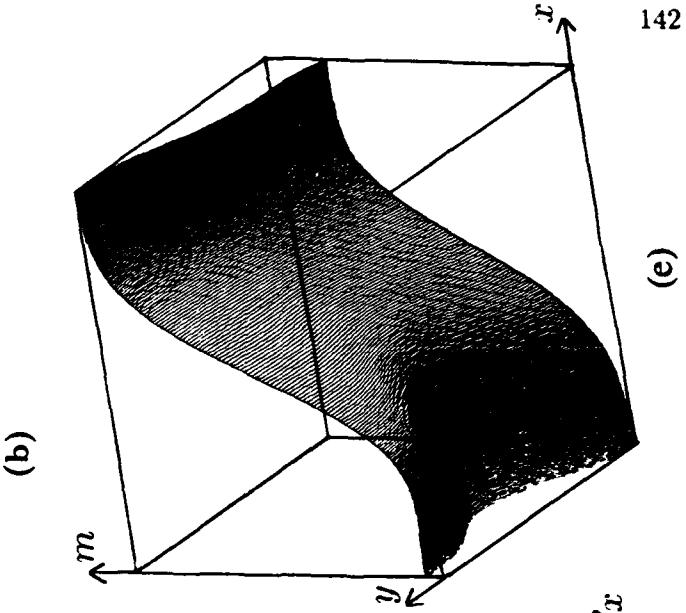
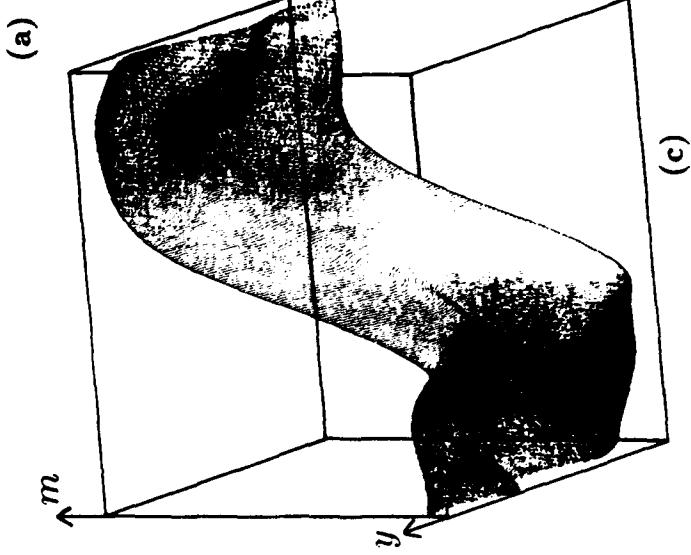
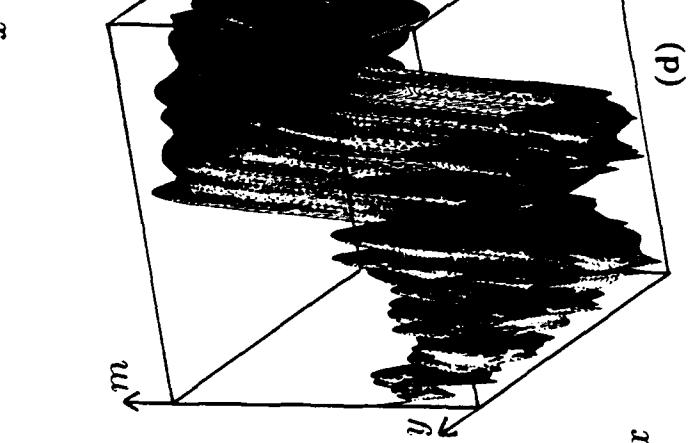
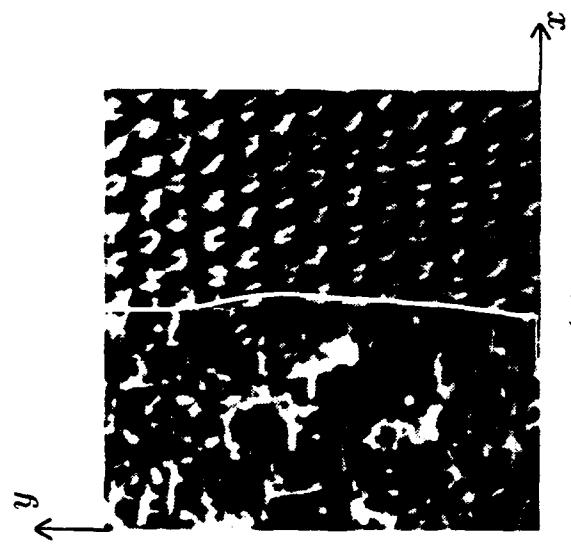
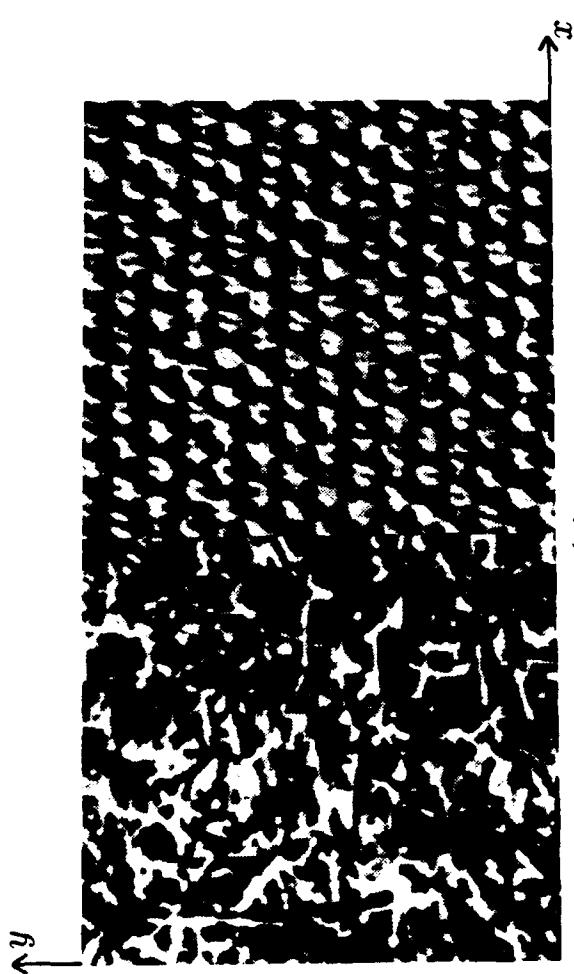
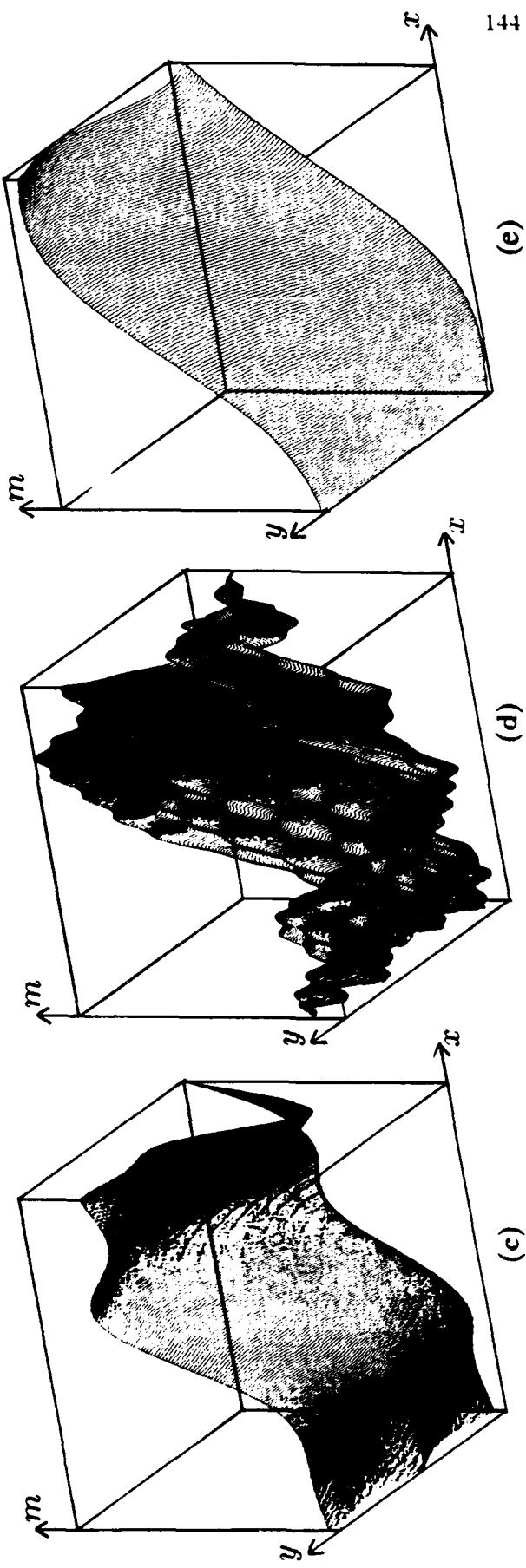
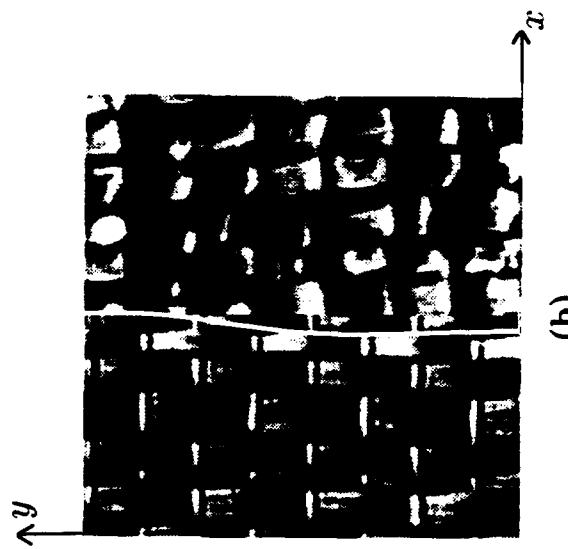
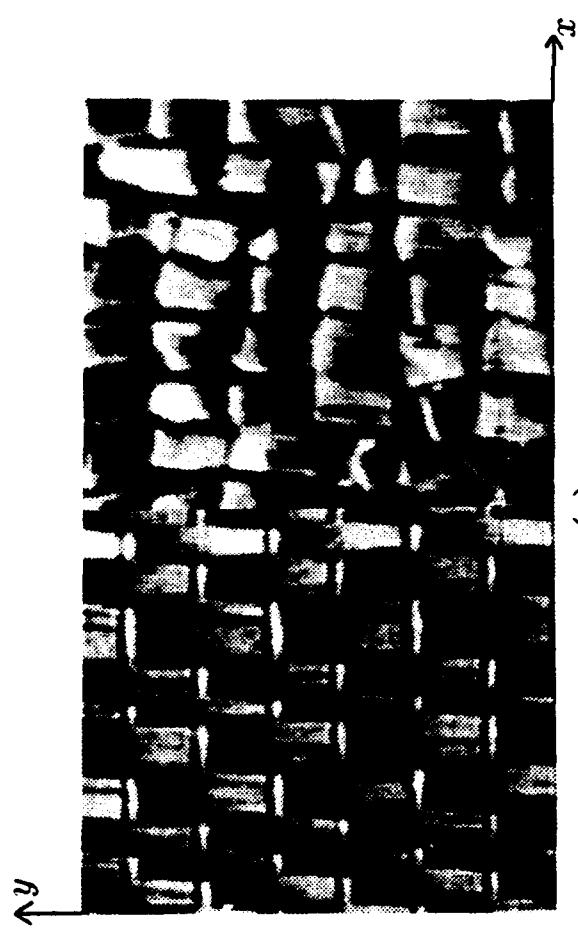


Fig. 8.11. Natural texture pair consisting of Brodatz's "straw matting" (D55) and "raffia" (D84):
(a) Input image.
(b) Canny edge detector applied to (c) and superimposed on (a).
(c) Gabor-filter output using a C1 filter, $\sigma = 36$ pixels, $F = 0.491$ cycles/pixel, $\phi = 59.4^\circ$, $P_E = 0.0001$. Maximum & minimum values for $m(x,y)$: 8.34×10^4 , 3.31×10^1 .
(d) Gabor-filter output $m(x,y)$, $\sigma = 24$ pixels, $F = 0.046$ cycle/pixel, $\phi = 87.4^\circ$, $P_E = 0.018$. Maximum & minimum values for $m(x,y)$: 4.39×10^6 , 1.66×10^3 .
(e) C2 second-stage output $m'(x,y)$ after filtering (d), $\sigma = 36$ pixels. Maximum & minimum values for $m(x,y)$: 9.36×10^{11} , 2.17×10^{11} .



to Fig. 8.11c. (In this case, edge localization is slightly better for the C1 output.)

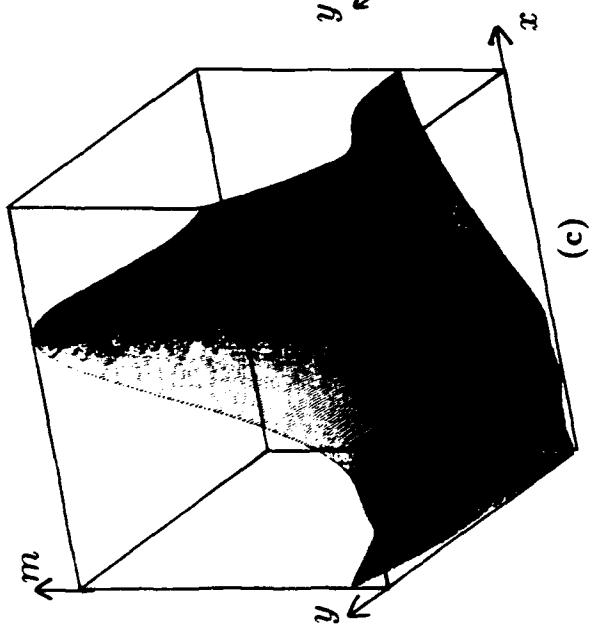
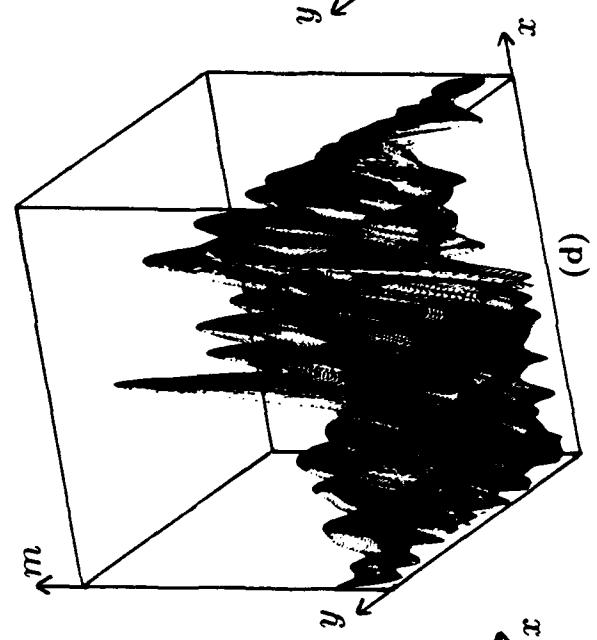
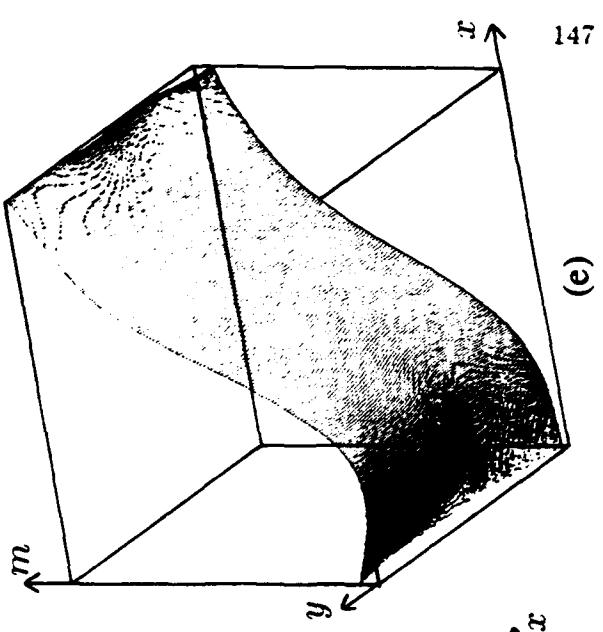
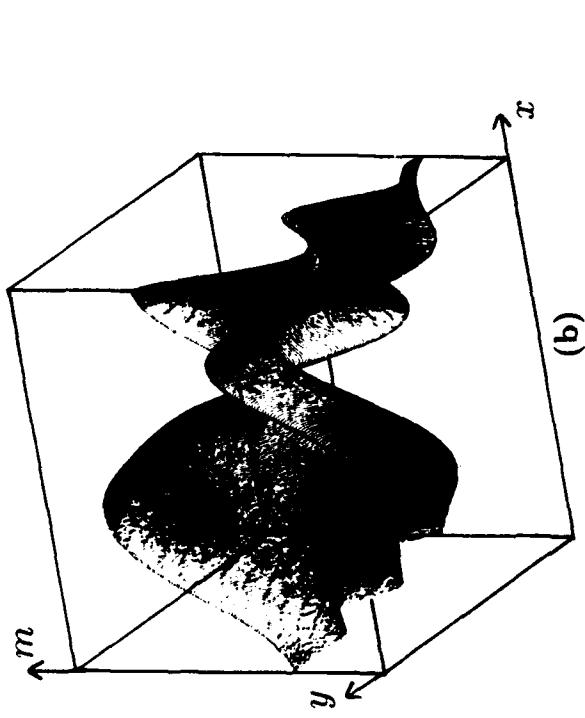
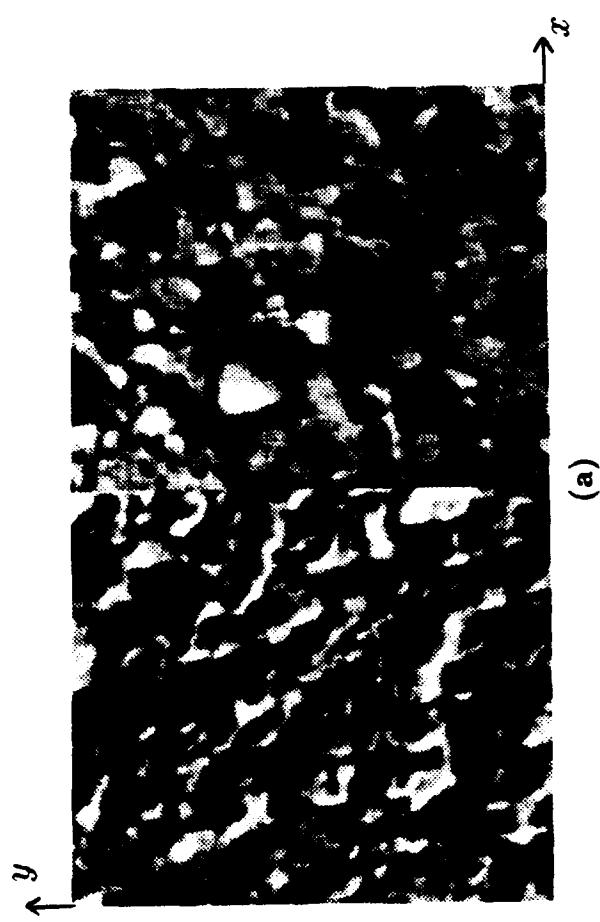
Fig. 8.12a consists of “pressed cork” (D4) and “beach sand” (D29). Rao classifies D4 as disordered and D29 as strongly-ordered [12]. In spite of this difference in type, the large variation in grain size of the sand makes these textures perceptually very similar. In this case, the values predicted by DFT and GFFS differ greatly even for $\sigma = 36$. Fig. 8.12b shows the result of applying a C1 filter to Fig. 8.12a with the DFT predicted frequency values. Note that although several peaks occur, there seems to be no indication of the texture-boundary location. A C2 filter was also applied to this texture pair with the same frequency value as in Fig. 8.12b, but with $\sigma = 12$. The result (not shown) shows little, if any, improvement. In this case, the DFT method fails to predict a useful center frequency. Also tested were the next two best frequency values predicted by the DFT method with similarly poor results (not shown). Fig. 8.12c shows the result of applying a C1 filter to Fig. 8.12a using the center frequency predicted by GFFS. Here, using a large value of σ produces too much discrimination within the D29 region. By reducing σ and applying a C2 configuration filter to this texture pair, the sequence shown in Figs. 8.12d and 8.12e was obtained. Note the high quality of the step after the second stage.

As the previous examples indicate, the DFT method can predict suitable filter parameters. It is also faster than GFFS. It is, however, deficient in the following ways:

- The DFT method does not always predict the same center frequency as GFFS. When this occurs, GFFS produces a better signature.

Fig. 8.12. Natural texture pair consisting of Brodatz's "pressed cork" (D4) and "beach sand" (D29):

- (a) Input image.
- (b) Gabor-filter output using a C1 filter with the DFT predicted value, $\sigma = 36$ pixels, $F = 0.002$ cycles/pixel, $\phi = 0.0^\circ$. Maximum & minimum values for $m(x, y)$: 6.32×10^7 , 5.41×10^7 .
- (c) Gabor-filter output using a C1 filter with values predicted by GFFS, $\sigma = 36$ pixels, $F = 0.474$ cycles/pixel, $\phi = 60.5^\circ$, $P_E = 0.0002$. Maximum & minimum values for $m(x, y)$: 2.53×10^5 , 1.07×10^2 .
- (d) Gabor-filter output using a C2 filter with values predicted by GFFS (first-stage output $m(x, y)$), $\sigma = 12$ pixels, $F = 0.478$ cycles/pixel, $\phi = -37.0^\circ$, $P_E = 0.191$. Maximum & minimum values for $m(x, y)$: 4.10×10^4 , 2.36×10^1 .
- (e) C2 second-stage output $m'(x, y)$ after filtering (d), $\sigma = 36$ pixels. Maximum & minimum values for $m(x, y)$: 6.37×10^9 , 1.87×10^9 .



- GFFS, with its quantitative measure of signature quality, provides a way of estimating the relative discriminability among different texture pairs. The DFT provides no such indication.
- In some cases, the best choice of center frequency depends on σ_x and σ_y . With GFFS, the best center frequency can be determined for any value of σ_x and σ_y . The DFT method, on the other hand, can produce only one value based on the size of the entire region.
- Section 7.2.5 described how zero padding can increase the number of center frequencies tested. In this way, it is possible to check that signature quality degrades gracefully as frequency varies. In practice, GFFS predicts similar signature quality over a wide range of adjacent frequency components. For the DFT method, the best choices often occur at very different frequency values. Thus it is uncertain how sensitive the “best” frequency is to slight frequency perturbations.
- The DFT method cannot predict center frequencies for the valley/ridge or difference-in-variance signatures.

8.2.7 Miscellaneous Texture Examples

The examples above are meant to typify Gabor-filter outputs, but there are exceptional cases. For example, if a filter is tuned to a frequency component that has similar magnitude in both textured regions, the output can be non-discriminating; i.e., the filter is not appropriate for discriminating between these two regions. If the regions

are uniform, then the filter output will be flat. If they are nonuniform, then the filter outputs may exhibit many fluctuations and show no distinguishing characteristics between regions. In the extreme case, when the frequency composition of the regions are similar, the image cannot be segmented. An example of such an image is the nonuniform texture pair consisting of R_s and mirror-image R_s presented in Fig. 2.3. Although many C1 and C2 configuration filters were applied to this image (results not shown), no distinct output signatures were found. It appears that the filter configurations developed here are ineffective in segmenting this texture. While this might be considered a limitation, at least it agrees with perception. It should be noted that because of the inherent variability within textures, no one algorithm can successfully segment all textures. For without a mathematical definition of texture, who is to say when a difference in input corresponds to a difference in texture or simply an acceptable variation within the texture? Thus, using human perception as a benchmark does not seem unreasonable.

Another exceptional case is when a filter is tuned to a frequency band not involved in determining a difference in texture. In this case, a discontinuity might occur at a location other than the texture boundary. This "problem" also exists for the human visual system in the form of optical illusions and the perception of structures within structures. Fig. 8.13 is an example demonstrating this phenomenon. Fig. 8.13a is an example of structure within texture (a similar example was mentioned in Chapter 2), first presented by Beck [9]. The left region consists of alternating rows of right-facing-Us and left-facing-Us, while the right region consists of alternating columns. The interesting feature here is that the texture boundary is not readily perceived. Rather, one tends to see three vertical black bars on the right. Beck refers to these bars as "emergent"

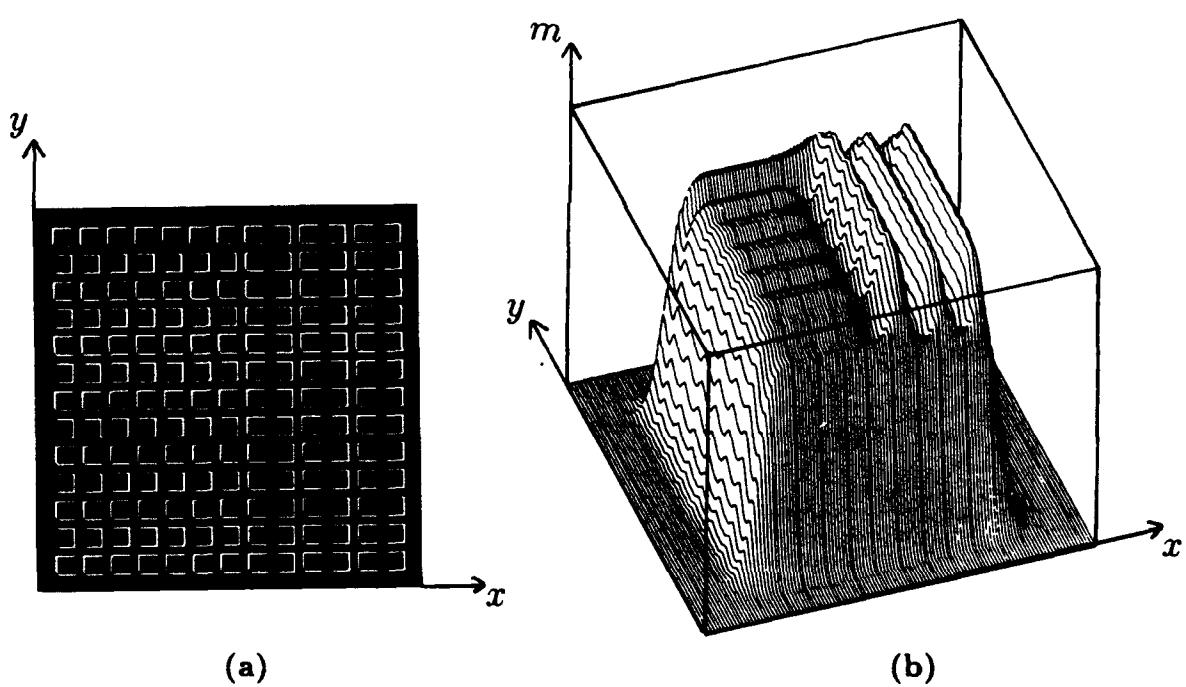


Fig. 8.13. Beck's emergent features:

(a) Input image.

(b) Gabor-filter output using a C1 filter, $\sigma = 1$ pixel, $F = 0.083$ cycles/pixel, $\phi = 0.0^\circ$.

features. Fig. 8.13b shows the Gabor-filter output for a C1 configuration filter (parameters determined by experiment). Note that the black bars are readily distinguished by the three vertical ridges in Fig. 8.13b, whereas the texture boundary produces no distinct output feature. This suggests that these simple filters are capable of detecting features previously thought to involve more complex processing (e.g., edge detection, feature linking, inhibitory interactions) [9, 56, 62].

Chapter 9

Conclusion

This thesis studies the design of filters for texture segmentation. It provides mathematical and experimental evidence suggesting that the application of Gabor filters to a textured image produces certain characteristic output signatures, that are useful for segmenting the image. Although the quantitative analysis is limited to a simplified texture subset, qualitative arguments and experimental results are provided indicating that the results apply in general.

Signature characteristics can best be described by dividing textures into two classes: uniform and nonuniform. For the class of *uniform* textures, output signatures occur in one of three forms, either a step, valley, or ridge. Analysis shows that the step signature occurs when two textured regions differ in texel-frequency composition. On the other hand, the valley and ridge signatures occur when two regions exhibit a texture-phase discontinuity. The regular nature of uniform textures produces smooth, well behaved signatures that are easy to segment using edge-detection methods. For the step signature, region-based techniques would also be effective. For *nonuniform* textures, the step, valley, and ridge signatures still occur; however, the presence of texel variation induces local fluctuations in the signatures. This makes subsequent segmentation less precise. For nonuniform textures, a fourth signature type can occur. This signature takes the form of a step change in average local output variation.

Although the characteristic signatures mentioned above are useful for texture segmentation, quality signatures occur only when the five filter parameters ($\sigma_x, \sigma_y, U, V, \theta$) are "tuned" to the texture being processed. Analysis shows that the choice of (σ_x, σ_y) depends on the texel spacing within the texture. This choice is a tradeoff between signature smoothness and accurate texture-boundary localization. It is further shown that when texel spacing differs in x and y , asymmetric filters (i.e., filters consisting of a non-circularly symmetric Gaussian) can be beneficial. For asymmetric filters, θ should be chosen to match the texel-spacing lattice. The choice of filter center frequency (U, V), on the other hand, is determined both by the texel spacing and by the difference in frequency content between texels in different regions. The texel spacing determines certain frequency harmonics in the texel Fourier transform. It is shown that, in order to avoid signature anomalies called overshoot and undershoot, (U, V) should equal one of these harmonics. It is further shown that the harmonic that differs most between texels in different regions produces the "best" signature (i.e., greatest amplitude difference). These frequency guidelines, however, were developed for uniform textures and are only approximately correct for nonuniform and natural textures. So, to provide effective frequency parameters for textures in general, an algorithm was developed that finds the "best" center frequencies for any given texture pair.

The filters used in this thesis are based on Gabor elementary functions. They were chosen because they have certain desirable properties for texture analysis. They are also suggested frequently in the literature. Results show, however, that it is the bandpass characteristic of these functions that is responsible for producing signatures; therefore, any class of functions that exhibit bandpass characteristics and are well localized in both

the space and spatial-frequency domains would suffice.

This thesis provides a detailed analysis of individual filters. There are, however, two questions that remain unresolved: Should an alternate nonlinearity be used after applying a bandpass filter? How should filters be configured to form a filter bank? The question of choosing a nonlinearity is discussed in Appendix A. The issue of filter bank configuration is briefly discussed below.

It is clear that in an autonomous texture-segmentation architecture, filters cannot be customized to individual textures. In principle, a bank of filters is required that span the expected orientation and frequency domain of the textures of interest. Although certain constraints on filter-bank design were presented in Chapter 6, two major issues remain:

1. What characteristics of the filter output should be used for segmentation - discontinuities at texture boundaries or texture-region information?
2. How should the response from multiple filters be integrated into a meaningful output - should one, possibly dominant, output be selected as representative (filter-output selection), or should many filter outputs be combined into a kind of feature vector (filter-output combination)?

This thesis shows that discontinuities in a single filter-output can be effective for texture segmentation; thus discontinuity detection and filter selection seem appropriate. In fact, for textures exhibiting only a texture-phase discontinuity as in Fig. 1.3, discontinuity detection is required, since the textured regions are identical. For texture classification, however, discontinuities alone cannot provide sufficient information. It seems that, for

this task, region data from multiple filters is required. In some cases, it seems that multiple filter outputs are also required for texture segmentation, especially if simulating human performance is desired. Consider the uniform texture pair in Fig. 3.3c. The human visual system has difficulty in segmenting this image; yet, many Gabor filters produce distinct filter-output discontinuities at the texture boundary. This is true, in fact, for any uniform texture pair whose regions have different constituent texels. This occurs because regions with different texels (and thus different texel Fourier transforms) produce different filter outputs. And, due to the lack of “noise” in uniform textures, even small filter-output differences are detectable. So why is Fig. 3.3c so difficult to segment for humans? Examining the Fourier transform magnitudes of the texels in Fig. 3.3c reveals that although occasional differences exist between the two transforms, on the average they are quite similar (much more so than for the texels of Fig. 3.3a). This suggests that the human visual system might be pooling information from multiple filters and basing segmentation on some form of average (possibly with thresholding). This procedure would also provide noise immunity. Clearly, both of these issues require additional study.

Appendix A

Choosing the Nonlinearity

This appendix further discusses the choice of nonlinearity for the Gabor filter (3.1). As shown below, computing the magnitude after filtering results in a loss of information. In particular, the phase component of the filter output is discarded. In fact, some experiments have shown that the image-phase component is more important in preserving image quality than is the amplitude component [81]. To avoid this information loss, some researchers have proposed methods for extracting phase information directly [53, 82].

The primary motivation for ignoring phase comes from psychophysical studies of the human visual system. Although neurophysiological evidence suggests that quadrature-pair filters might exist in the visual cortex [83] (thus enabling phase detection), certain psychophysical results suggest that humans do not encode phase information directly [84], at least not for texture segmentation. Consequently, some researchers have explicitly eliminated phase information from their texture-segmentation algorithms [38, 56]. Although, admittedly, information is lost by ignoring phase, some phase-related phenomena can be recovered directly from the amplitude envelope (see Bovik [66] for a discussion of the effect of phase on the amplitude envelope). This is because the phase and amplitude components are not independent: a change in one will produce a change in the other.

To further explore these points, suppose that instead of computing the magnitude of the GEF-filtered image, one simply demodulates this image and applies a lowpass filter. Demodulating the GEF-filtered image (5.17), essentially eliminates the complex exponential leaving a pair of offset gates with *complex* coefficients T_1 and T_2 . Contrast this with (5.18), where the magnitude operation has been applied, and only the magnitudes of T_1 and T_2 determine the gate amplitudes. If T_1 and T_2 differ only in sign, the Gabor-filter output m in (5.18) is nondiscriminating. This implies that if textures differ only in the sign of contrast, the textures cannot be discriminated. This is precisely the argument used by Malik and Perona in criticizing the magnitude computation [56]. By using the demodulation approach, though, the coefficients T_1 and T_2 after demodulation are complex. Hence the filter output will reflect not only differences in sign, but also differences in phase. Although this approach is more discriminating, the sensitivity to phase will lead to segmentations that are not consistent with human performance.

There is a method for retaining sign differences between T_1 and T_2 while ignoring phase information. The method involves convolving the image with only the real portion of a GEF. More precisely, define a new filter h_r as a Gaussian modulated sinusoid (cf. (3.2))

$$h_r(x, y) = g(x'y') \cos[(Ux + Vy) + \phi] \quad (\text{A.1})$$

where ϕ is some arbitrary constant phase angle. Let H_r be the Fourier transform of h_r .

$$H_r(u, v) = 1/2(H_+e^{-j\phi} + H_-e^{j\phi}) \quad (\text{A.2})$$

where

$$\begin{aligned} H_+ &= \exp \left\{ -\frac{1}{2} [(\sigma_x[u+U]')^2 + (\sigma_y[v+V]')^2] \right\} \\ H_- &= \exp \left\{ -\frac{1}{2} [(\sigma_x[u-U]')^2 + (\sigma_y[v-V]')^2] \right\} \end{aligned}$$

Note that H_r is similar to (3.4) except that it is symmetric about the frequency origin.

Applying H_r to I in (5.9) gives (cf. (5.14-5.15))

$$I_r(u, v) \approx \frac{\pi}{\Delta x \Delta y} (I_{r+} + I_{r-}) \quad (\text{A.3})$$

where

$$\begin{aligned} I_{r+} &= H_+ e^{-j\phi} S(u+U, v+V) \left\{ T_1(U, V) + T_2(U, V) e^{-jr(u+U)} \right\} \\ I_{r-} &= H_- e^{j\phi} S(u-U, v-V) \left\{ T_1(-U, -V) + T_2(-U, -V) e^{-jr(u-U)} \right\} \end{aligned}$$

Note that I_r is equivalent to the sum of I_f in (5.14) and the mirror image of I_f .

We now demodulate and lowpass filter I_r . This is equivalent to shifting both I_f and the mirror image of I_f to the origin and summing them. This results in

$$I_d(u, v) \approx K \left\{ T_1(U, V) + T_2(U, V) e^{-jr u} \right\} + K \left\{ T_1(-U, -V) + T_2(-U, -V) e^{-jr u} \right\} \quad (\text{A.4})$$

where

$$K = \frac{\pi}{\Delta x \Delta y} \exp \left\{ -\frac{1}{2} [(\sigma_x u')^2 + (\sigma_y v')^2] \right\} S(u, v)$$

For real images $T_1(U, V)$ and $T_1(-U, -V)$ are complex conjugates, as are $T_2(U, V)$ and $T_2(-U, -V)$. Therefore, (A.4) reduces to

$$I_d(u, v) \approx 2K \left\{ \operatorname{Re}[T_1] + \operatorname{Re}[T_2]e^{-jru} \right\} \quad (\text{A.5})$$

By arguments similar to those used for deriving the step signature in Section 5.2.1, the final output of the alternate filter $i_d(x, y)$, which is given by the inverse Fourier transform of (A.5), approximates two offset gate functions coincident with the region boundaries. The amplitudes of the gates are proportional to the real parts of T_1 and T_2 rather than their magnitude. Thus *the sign is preserved without retaining phase information*. This approach then can discriminate textures whose texels differ only in the sign of contrast.

It is important to point out that all of the operations used here are linear, and as mentioned in Chapter 3, some form of nonlinearity is essential in simulating human performance (assuming this is desirable). If a suitable nonlinearity could be found and imposed after demodulation, this method could provide an alternative to the more elaborate architecture proposed by Malik and Perona [56].

Farrokhnia and Jain used h_r in their texture-segmentation work, but still employed the magnitude computation [69, 70]; however, it was just shown that the sign information is still lost with this approach. The contribution here is analytically showing the potential of demodulating the filter output and contrasting this approach with other methods. It is important to realize that simply using h_r as a filter does not guarantee that sign information will be preserved.

Appendix B

Overshoot and Undershoot

Section 5.2.1 considered what happens when a Gabor filter is applied to a uniformly textured image that contains two textures whose respective texels differ. As was shown, if the Gabor filter is not tuned to an harmonic, the step signature can exhibit overshoot and/or undershoot near the texture boundary. Also, if the Gabor filter is tuned to an harmonic, overshoot *cannot* occur in the step signature. This appendix discusses the issues of overshoot and undershoot.

First, consider the case when the Gabor filter is not tuned to an harmonic. Consider the expression for i_f in (5.17). This represents the output of a GEF-filtered image. If the GEF is not tuned to an harmonic, then by the ridge-signature arguments of Section 5.2.2, $s_f(x, y)$ and $s_f(x - r, y)$ are complex and defined as in (5.31) and (5.32). Computing the complex magnitude of (5.17) for this case gives

$$m(x, y) = |i_f(x, y)| = \frac{2\pi}{\Delta x \Delta y} \sqrt{P_r P_r^* + P_0 P_0^* + P_r P_0^* + P_0 P_r^*} \quad (B.1)$$

where $P_0 = T_1(U, V)s_f(x, y)$ and $P_r = T_2(U, V)s_f(x - r, y)$. By an analysis similar to that for the ridge, it can be shown that

$$m(x, y) \approx \hat{A}_1 = \frac{2\pi}{\Delta x \Delta y} |T_1| \gamma_1$$

far to the left of the texture boundary, and

$$m(x, y) \approx \hat{A}_2 = \frac{2\pi}{\Delta x \Delta y} |T_2| \gamma_1$$

far to the right of the texture boundary, where γ_1 is defined in (5.37). Per (5.34), b_r in (5.37) is a cosine-modulated Gaussian. Then, for large r , $\gamma_1 < 1$. Thus \hat{A}_1 and \hat{A}_2 are less than when the filter is tuned to an harmonic (cf. A_1 and A_2 at the end of Section 5.2.1). Because the imaginary component of s_f does not go to zero near the texture boundary, however, constructive and destructive interference can occur between terms in (B.1) to produce values of $m < \min(\hat{A}_1, \hat{A}_2)$ (undershoot) or $> \max(\hat{A}_1, \hat{A}_2)$ (overshoot).

An analytical demonstration of overshoot or undershoot using (B.1) is difficult. The problem is that the location of the overshoot/undershoot is not in general at the texture boundary, and analyzing (B.1) away from the boundary is complicated by the interactions of several complex variables. As an alternative, Chapter 8 presents examples of Gabor-filtered texture pairs that exhibit overshoot and undershoot and compares the corresponding outputs to the signatures predicted by (B.1).

We now show for the scenario of Section 5.2.1 that if a Gabor filter is tuned to an harmonic, overshoot cannot occur in a step signature. The absence of overshoot can be demonstrated by showing that (5.18) cannot achieve a value $|i_f| > \left(\frac{2\pi}{\Delta x \Delta y}\right) \cdot \max(|T_1|, |T_2|)$; i.e., $|i_f|$ cannot exceed its maximum asymptotic value.

Squaring (5.18), eliminating the constants Δx and Δy , and realizing that $s_f(x -$

$r, y) \approx 1 - s_f(x, y)$, the problem reduces to showing that $k \leq \max(|T_1|^2, |T_2|^2)$, where

$$k = T_1 T_1^* s_f^2 + T_2 T_2^* (1 - s_f)^2 + (T_1^* T_2 + T_1 T_2^*) s_f (1 - s_f).$$

Since k is quadratic in s_f , it has one maximum or minimum. The quantity

$$\frac{d^2 k}{ds^2} = T_1 T_1^* + T_2 T_2^* - T_1 T_2^* - T_1^* T_2 = (T_1 - T_2)(T_1 - T_2)^*$$

is always non-negative. Thus, k has a single minimum, and its maximum values must occur at the end points; i.e., at $s_f = 0$ or $s_f = 1$. If $s_f = 1$, then $k = |T_1|^2$. If $s_f = 0$, then $k = |T_2|^2$, which implies that $k \leq \max(|T_1|^2, |T_2|^2)$.

Appendix C

Estimating Ridge Height

As Section 5.2.2 showed, if an improperly tuned Gabor filter is applied to an image exhibiting a texture-phase difference between two regions, a *ridge signature* can arise at the region (texture) boundary. To segment such an image, the location of the ridge must be detected. The ease of detecting a ridge signature is related to its height. This appendix develops a method for computing this height.

Ridge height at a texture-phase discontinuity is given by $\gamma_1 C z(\alpha, \psi)$ in (5.40), where $\gamma_1 C$ represents the Gabor-filtered output far removed from the discontinuity and $z > 1$ represents the relative ridge height. To determine z , α is computed by resolving (5.37), (5.38), and (5.41). Assume that the spatial extent of the GEF h is effectively contained within the image boundaries. Thus, (5.37) and (5.38) become

$$\begin{aligned} c_1 &\approx 2 \int_{-\infty}^{\infty} \int_0^{\infty} h_r(\alpha, \beta) d\alpha d\beta \\ &\approx 2 \int_{-\infty}^{\infty} \int_0^{\infty} g(x', y') \cos[\delta U x + \delta V y] dx dy \\ c_2 &\approx 2 \int_{-\infty}^{\infty} \int_0^{\infty} h_i(\alpha, \beta) d\alpha d\beta \\ &\approx 2 \int_{-\infty}^{\infty} \int_0^{\infty} g(x', y') \sin[\delta U x + \delta V y] dx dy \end{aligned}$$

After expanding the trigonometric functions and separating the integrals, we find that

$$c_1 = 4c_{\delta U}c_{\delta V}$$

$$c_2 = 4c_{\delta V} s_{\delta U}$$

where

$$c_{\delta U} = \frac{1}{\sqrt{2\pi}\sigma_x} \int_0^\infty e^{-1/2(x/\sigma_x)^2} \cos(\delta U x) \quad (C.1)$$

$$c_{\delta V} = \frac{1}{\sqrt{2\pi}\sigma_y} \int_0^\infty e^{-1/2(y/\sigma_y)^2} \cos(\delta V y) \quad (C.2)$$

$$s_{\delta U} = \frac{1}{\sqrt{2\pi}\sigma_x} \int_0^\infty e^{-1/2(x/\sigma_x)^2} \sin(\delta U x) \quad (C.2)$$

Thus, α becomes

$$\alpha = c_2/c_1 = s_{\delta U}/c_{\delta U}. \quad (C.3)$$

Evaluation of (C.1-C.3) gives

$$\alpha = \sqrt{2/\pi} \delta U \cdot \sigma_x F_{1,1} \left(\frac{1}{2}; \frac{3}{2}; \frac{1}{2}(\delta U \cdot \sigma_x)^2 \right) \quad (C.4)$$

where $F_{1,1}$ represents the Kummer's confluent hypergeometric series. Thus, α depends only on $\delta U \cdot \sigma_x$. Table C.1 shows values of α as a function of $\delta U \cdot \sigma_x$

Given a Gabor filter with parameters $(\sigma_x, \sigma_y, U, V, \theta)$, we can determine δU by finding the harmonic indices (\hat{k}, \hat{l}) such that $|U - 2\pi\hat{k}/\Delta x|$ and $|V - 2\pi\hat{l}/\Delta y|$ are minimized. Then, $\delta U = U - 2\pi\hat{k}/\Delta x$, and α can be determined from Table C.1 or (C.4). Given α and using (\hat{k}, \hat{l}) to compute ψ in (5.26), the relative ridge height can be determined from (5.42).

Table C.1. Numerical evaluation of the integral ratio $\alpha = s_{\delta U}/c_{\delta U}$ as a function of the product of parameters δU and σ_x .

$\delta U \cdot \sigma_x$	α
0.05	0.255
0.0937	0.499
0.10	0.536
0.20	1.343
0.288	2.834
0.30	3.166
0.50	40.96

Appendix D

Implementation Details

This appendix discusses the implementation details of filter application and edge detection. Applying a Gabor filter to an image involves convolving the image with a GEF and then computing the magnitude of the convolution result. Convolution is performed in the frequency domain by using the Discrete Fourier Transform (DFT). The steps in this procedure are summarized below:

1. Input the filter parameters λ , f , ϕ , and σ , where λ defines the aspect ratio of the filter, f and ϕ determine the filter center frequency, and σ corresponds to σ_x in (3.3).
2. Input the textured image I as a square array ($n \times n$).
3. Define the dimensions of the filter in x and y as $x = 6\sigma + 1$ and $y = 6\lambda\sigma + 1$ respectively.
4. Append zeros to the input array I to form a square array I_d of dimension $d = n + \max(x, y)$.
5. Compute the DFT of the expanded input array I_d .
6. Construct an $x \times y$ filter array F by sampling the equation for a GEF determined by the filter parameters input at step (1).

7. Append zeros to the filter array F to form a square array F_d equal in size to the expanded input array I_d .
8. Compute the DFT of the expanded filter array F_d .
9. Multiply the DFT of I_d by the DFT of F_d , and compute the inverse DFT of the result. Call the resulting output array R_d . At this point the GEF has been applied to the input image. If a lowpass filter is to be used to smooth the output (as for a C2 configuration filter), the following additional steps are performed:
 - Discard all points within $\max(x, y)$ from the boundaries of R_d to eliminate boundary anomalies. Call the new array R_m , where $m = d - 2 \max(x, y)$ is its dimension.
 - Input the size parameter σ_l for the lowpass filter.
 - Define both the x and y dimensions of the lowpass filter equal to $s = 6\sigma_l + 1$.
 - Construct an $s \times s$ lowpass filter array G by sampling the equation for a circularly symmetric Gaussian with parameter σ_l .
 - Append zeros to R_m to form a square array R_p of dimension $p = m + s$.
 - Append zeros to G to form a square array G_p equal in size to R_p .
 - Compute the DFT of R_p and the DFT of G_p .
 - Multiply the DFT of R_p by the DFT of G_p , and compute the inverse DFT of the result. Call the result S_p . At this point the lowpass filter has been applied to GEF output.
10. Compute the magnitude of the output array (either R_d or S_p).

11. Discard appropriate boundary points.

12. Scale the result to the range [0 – 255].

Note that the DFT algorithm used in this implementation omits the normal $1/N^2$ scaling factor associated with the DFT. This results in abnormally large output values when DFTs are multiplied and is the reason why the reported filter output values are so large.

The detection of edges is implemented as described by Canny [75]. Given the filter output as computed above, the directional derivatives in both the x and y directions are computed at each point (i, j) in the output. The nearest 2 neighbors in each direction are used to estimate these directional derivatives. Next, the gradient is estimated from the directional derivatives at each point (i, j) . These gradients are then nonmaximally suppressed. That is, at each point (i, j) , the gradient is compared to gradients in a eight-point neighborhood. If the gradient at point (i, j) is not the local maximum, it is discarded. Finally, thresholding is used to eliminate small, spurious gradients. The remaining nonzero points correspond to the edges in the filter output.

Bibliography

- [1] D. Dunn, W. Higgins, and J. Wakeley, "2-D analysis of Gabor-filter output signatures for texture segmentation," in *IEEE Int. Conf. Acoustics, Speech, and Signal Proc.*, March 1992.
- [2] D. Dunn, W. Higgins, A. Maida, and J. Wakeley, "Texture boundary classification using Gabor elementary functions." *7th IEEE Workshop Multidimen. Signal Processing*, Lake Placid, NY, September 1991.
- [3] D. Dunn, W. Higgins, A. Maida, and J. Wakeley, "Texture boundary classification using Gabor elementary functions," in *SPIE Proc. Visual Commun. Image Processing '91: Sixth in a Series*, vol. 1606, pp. 541-552, November 1991.
- [4] D. Dunn, W. Higgins, and J. Wakeley, "Determining Gabor-filter parameters for texture segmentation." submitted *SPIE Conf. Intelligent Robots and Visual Commun.*, 1992.
- [5] D. Dunn, W. Higgins, and J. Wakeley, "Texture segmentation using 2-D Gabor elementary functions." submitted *IEEE Trans. Pattern Anal. Machine Intell.*, 1992.
- [6] S. B. Flexner and L. C. Hauck, eds., *The Random House Dictionary of the English Language*. New York, NY: Random House, second ed., 1987. Unabridged.
- [7] D. Blostein and N. Ahuja, "Shape from texture: Integrating texture-element extraction and surface estimation," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 11, pp. 1233-1251, December 1989.

- [8] H. C. Nothdurft, "Sensitivity for structure gradient in texture discrimination tasks," *Vision Research*, vol. 25, no. 12, pp. 1957–1968, 1985.
- [9] J. Beck, K. Prazdny, and A. Rosenfeld, "A theory of textural segmentation," in *Human and Machine Vision* (J. Beck, B. Hope, and A. Rosenfeld, eds.), pp. 1–38, Academic Press, 1983.
- [10] B. Julesz, "Visual pattern discrimination," *IRE Trans. Inf. Theory*, vol. 8, pp. 84–92, 1962.
- [11] R. Gurnsey and R. Browse, "Micropattern properties and presentation conditions influencing visual texture discrimination," *Percept. Psychophys.*, vol. 41, pp. 239–252, 1987.
- [12] A. R. Rao, *A Taxonomy for Texture Description and Identification*. New York, NY: Springer-Verlag, 1990.
- [13] P. Brodatz, *Textures: A Photographic Album for Artists and Designers*. New York, NY: Dover, 1966.
- [14] W. K. Pratt, *Digital Image Processing*. New York, NY: Wiley, second ed., 1991.
- [15] B. J. Kröse, "Local structure analyzers as determinants of preattentive pattern discrimination," *Biol. Cybern.*, vol. 55, pp. 289–298, 1987.
- [16] P. Kube, "Polynomial shift-invariant operators for texture segmentation," *Proc. IEEE Comp. Vision Pattern Recog.*, pp. 100–104, 1988.

- [17] R. Conners and C. Harlow, "A theoretical comparison of texture algorithms," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-2, pp. 204-222, May 1980.
- [18] A. Gagolowicz, "A new method for texture fields synthesis: Some applications to the study of human vision," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-3, pp. 520-533, September 1981.
- [19] J. W. Woods, "Two-dimensional discrete Markov random fields," *IEEE Trans. Information Theory*, vol. IT-18, pp. 232-240, March 1972.
- [20] J. M. Francos and A. Z. Meiri, "A compound Poisson-cliques random field model for texture singularities," in *Proc. Int. Conf. Acoustics, Speech, Sig. Proc.*, pp. 2669-2672, 1991.
- [21] B. J. Super and A. C. Bovik, "Localized measurement of image fractal dimension using Gabor filters," *Journal Visual Commun. Image Rep.*, vol. 2, no. 2, pp. 114-128, 1991.
- [22] A. Pentland, "Fractal-based description of natural scenes," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-6, pp. 661-674, November 1984.
- [23] B. Julesz, "A theory of preattentive texture discrimination based on first-order statistics of textons," *Biol. Cybern.*, vol. 41, pp. 131-138, 1981.
- [24] B. Julesz, "Textons, the elements of texture perception, and their interactions," *Nature*, vol. 290, pp. 91-97, March 1981.

- [25] D. Marr, *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*. San Francisco, CA: W. H. Freeman and Company, 1982.
- [26] N. Ahuja and A. Rosenfeld, "Mosaic models for textures," *IEEE Trans. Pattern Anal. Machine Intell*, vol. PAMI-3, pp. 225-235, January 1981.
- [27] A. P. Pentland, "Perceptual organization and the representation of natural form," *Artificial Intelligence*, vol. 28, pp. 293-331, May 1986.
- [28] A. Treisman, "Preattentive processing in vision," *Computer Vision, Graphics, and Image Proc.*, vol. 31, pp. 156-177, 1985.
- [29] B. Julesz, E. N. Gilbert, and L. A. Shepp, "Inability of humans to discriminate between visual textures that agree in second-order statistics-revisited," *Perception*, vol. 2, pp. 391-405, 1973.
- [30] U. Neisser, *Cognitive Psychology*. New York, NY: Appleton-Century-Crofts, 1967.
- [31] B. Julesz, "Texton gradients: The texton theory revisited," *Biol. Cybern.*, vol. 54, pp. 245-251, 1986.
- [32] J. R. Bergen and E. H. Adelson, "Early vision and texture perception," *Nature*, vol. 333, pp. 363-364, May 1988.
- [33] B. J. Kröse, *A description of visual structure*. PhD thesis, Delft University of Technology, Delft, The Netherlands, 1986.

- [34] J. R. Pomerantz, "Perceptual organization in information processing tasks," in *Perceptual Organization* (M. Kubovy and J. R. Pomerantz, eds.), pp. 141–181, Hillsdale, NJ: Erlbaum, 1981.
- [35] R. Sekuler and R. Blake, *Perception*. New York, NY: Alfred A. Knoff, 1985.
- [36] D. H. Hubel and T. N. Wiesel, "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex," *J. Physiol. (London)*, vol. 160, pp. 106–154, 1962.
- [37] D. Marr and E. Hildreth, "Theory of edge detection," *Proc. R. Soc. Lond. B*, vol. 207, pp. 187–217, 1980.
- [38] K. Iwama and A. S. Maida, "Organizing and integrating edge segments for texture discrimination," *J. Expt. Theor. Artif. Intell.*, vol. 2, pp. 113–132, 1989.
- [39] F. W. Campbell and J. G. Robson, "Application of Fourier analysis to the visibility of gratings," *J. Physiol (London)*, vol. 197, pp. 551–566, 1968.
- [40] K. K. DeValois, R. L. DeValois, and E. W. Yund, "Responses of striate cortex cells to grating and checkerboard patterns," *J. Physiol.*, vol. 291, pp. 483–505, 1979.
- [41] R. L. DeValois, "Early visual processing: Feature detection or spatial filtering?," in *Recognition of Pattern and Form* (S. Levin, ed.), pp. 152–174, Berlin: Springer-Verlag, 1982.
- [42] J. J. Kulikowski and P. O. Bishop, "Fourier analysis and spatial representation in the visual cortex," *Experientia*, vol. 37, pp. 160–163, 1981.

- [43] J. J. Kulikowski and P. O. Bishop, "Silent periodic cells in the cat striate cortex," *Vision Research*, vol. 22, pp. 191-200, 1982.
- [44] J. J. Kulikowski, S. Marcelja, and P. Bishop, "Theory of spatial position and spatial frequency relations in the receptive fields of simple cells in the visual cortex," *Biol. Cybern.*, vol. 43, pp. 187-198, 1982.
- [45] L. Maffei, C. Morrone, M. Pirchio, and G. Sandini, "Responses of visual cortical cells to periodic and non-periodic stimuli," *J. Physiology*, vol. 296, pp. 27-47, 1979.
- [46] S. Marcelja, "Mathematical description of the responses of simple cortical cells," *J. Opt. Soc. of Amer.*, vol. 70, no. 11, pp. 1297-1300, 1980.
- [47] B. Julesz and T. Caelli, "On the limits of Fourier decomposition in visual texture perception," *Perception*, vol. 8, pp. 69-73, 1979.
- [48] D. M. MacKay, "Strife over visual cortical function," *Nature*, vol. 289, pp. 117-118, January 1981.
- [49] D. Gabor, "Theory of communication," *J. IEE (London)*, vol. 93, pp. 429-457, 1946.
- [50] J. Daugman, "Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters," *J. Opt. Soc. Am. A*, vol. 2, pp. 1160-1169, July 1985.
- [51] J. Daugman, "Two-dimensional spectral analysis of cortical receptive field profiles," *Vision Research*, vol. 20, pp. 847-856, 1980.

- [52] J. Daugman, "Spatial visual channels in the Fourier plane," *Vision Research*, vol. 24, no. 9, pp. 891–910, 1984.
- [53] A. Bovik, M. Clark, and W. Geisler, "Multichannel texture analysis using localized spatial filters," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 12, pp. 55–73, January 1990.
- [54] J. Daugman, "Complete discrete 2-D Gabor transforms by neural networks for image analysis and compression," *IEEE Trans. Acoustics, Speech, and Signal Processing*, vol. 36, no. 7, pp. 1169–1179, 1988.
- [55] I. Fogel and D. Sagi, "Gabor filters as texture discriminator," *Biol. Cybern.*, vol. 61, pp. 103–113, 1989.
- [56] J. Malik and P. Perona, "Preattentive texture discrimination with early vision mechanisms," *J. Opt. Soc. Amer. A*, vol. 7, pp. 923–932, May 1990.
- [57] S. Mallat, "A theory for multiresolution signal decomposition: The wavelet representation," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 11, pp. 674–693, July 1989.
- [58] M. Porat and Y. Zeevi, "Localized texture processing in vision: Analysis and synthesis in the Gaborian space," *IEEE Trans. Biomedical Eng.*, vol. 36, pp. 115–129, January 1989.
- [59] T. T. Reed and H. Wechsler, "Segmentation of textured images and Gestalt organization using spatial/spatial-frequency representations," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 12, pp. 1–12, January 1990.

- [60] M. Turner, "Texture discrimination by Gabor functions," *Biol. Cybern.*, vol. 55, pp. 71-82, 1986.
- [61] C. Heil and D. Walnut, "Continuous and discrete wavelet transforms," *SIAM Review*, vol. 31, pp. 628-666, December 1989.
- [62] B. Manjunath and R. Chellappa, "A computational approach to boundary detection," in *IEEE Conf. Comp. Vision Pattern Rec.*, pp. 358-363, 1991.
- [63] B. J. Super and A. C. Bovik, "Three-dimensional orientation from texture using Gabor wavelets," in *Proc. SPIE Conf. of Visual Communications and Image Proc.*, vol. 1606, pp. 574-586, 1991.
- [64] R. Young, "The Gaussian derivative theory of spatial vision: Analysis of cortical cell receptive field line-weighting profiles," Tech. Rep. GMR-4920, General Motors Research Laboratories, May 1987.
- [65] R. N. Bracewell, *The Fourier Transform and Its Applications*. New York: McGraw-Hill, 1978.
- [66] A. C. Bovik, "Analysis of multichannel narrow-band filters for image texture segmentation," *IEEE Trans. Signal Proc.*, vol. 39, no. 9, pp. 2025-2043, 1991.
- [67] B. Julesz. Personal communication cited in Malik and Perona [56].
- [68] M. Clark and A. Bovik, "Experiments in segmenting texton patterns using localized spatial filters," *Pattern Recognition*, vol. 22, no. 6, pp. 707-717, 1989.

- [69] F. Farrokhnia and A. K. Jain, "A multi-channel filtering approach to texture segmentation," in *IEEE Conf. Comp. Vision Pattern Rec.*, pp. 364-370, 1991.
- [70] A. K. Jain and F. Farrokhnia, "Unsupervised texture segmentation using Gabor filters," *Pattern Recognition*, vol. 23, 1991.
- [71] J. K. Hawkins, "Textural properties for pattern recognition," in *Picture Processing and Psychopictorics* (B. C. Lipkin and A. Rosenfeld, eds.), pp. 347-370, New York, NY: Academic Press, 1970.
- [72] J. Sklansky, "Image segmentation and feature extraction," *IEEE Trans. Syst. Man Cybern.*, vol. 8, pp. 237-247, 1978.
- [73] M. Porat and Y. Zeevi, "The generalized Gabor scheme of image representation in biological and machine vision," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 10, pp. 452-467, July 1988.
- [74] H. Wilson and J. Bergen, "A four mechanism model for threshold spatial vision," *Vision Research*, vol. 19, pp. 19-32, 1979.
- [75] J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-8, pp. 679-698, November 1986.
- [76] H. Urkowitz, *Signal Theory and Random Processes*. Dedham, Mass.: Artech House, Inc., 1983.
- [77] A. Oppenheim and R. Schafer, *Discrete-time Signal Processing*. Englewood Cliffs, NJ: Prentice Hall, 1989.

- [78] I. Daubechies, "The wavelet transform, time-frequency localization and signal analysis," *IEEE Trans. Inform. Theory*, vol. 36, pp. 961-1005, September 1990.
- [79] L. Cohen, "Time-frequency distributions - a review," *Proc. IEEE*, vol. 77, no. 7, pp. 941-981, 1989.
- [80] H. Stark and J. Woods, *Probability, Random Processes, and Estimation Theory for Engineers*. Englewood Cliffs, NJ: Prentice Hall, 1986.
- [81] A. Oppenheim and J. Lim, "The importance of phase in signals," *Proc. of the IEEE*, vol. 69, pp. 529-541, May 1981.
- [82] J. duBuf and P. Heitkämper, "Texture features based on Gabor phase," *Signal Processing*, vol. 23, pp. 227-244, 1991.
- [83] D. A. Pollen and S. F. Ronner, "Spatial computation performed by simple and complex cells in the visual cortex of the cat," *Vision Research*, vol. 22, pp. 101-118, 1982.
- [84] I. Rentschler, M. Hubner, and T. Caelli, "On the discrimination of compound Gabor signals and textures," *Vision Research*, vol. 28, no. 2, pp. 279-291, 1988.